Trash Classification Using Deep Neural Network

by

Dhrubajyoti Das

Roll No: 17MCSE007P

A thesis submitted for the partial fulfillment of the requirement for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering(CSE)

CHITTAGONG UNIVERSITY OF ENGINEERING & TECHNOLOGY

Chattogram-4349, Bangladesh

November, 2023

CERTIFICATION

The thesis titled **"Trash Classification Using Deep Neural Network"** submitted by **Dhrubajyoti Das**, Roll No **17MCSE007P**, Session **2017-2018** has been accepted as satisfactory in partial fulfillment of the requirement for the degree of MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING on 28/11/2023

BOARD OF EXAMINER

1. _____ Dr. Kaushik Deb Professor Department of Computer Science and Engineering Chittagong University of Engineering & Technology Chattogram- 4349, Bangladesh.	Chairman (Supervisor)
2. _____ Dr. Abu Hasnat Mohammad Ashfak Habib Professor & Head Department of Computer Science and Engineering Chittagong University of Engineering & Technology Chattogram- 4349, Bangladesh.	Member (Ex-Officio)
3. _____ Dr. Muhammad Ibrahim Khan Professor Department of Computer Science and Engineering Chittagong University of Engineering & Technology Chattogram- 4349, Bangladesh.	Member
4. _____ Dr. Asaduzzaman Professor Department of Computer Science and Engineering Chittagong University of Engineering & Technology Chattogram- 4349, Bangladesh.	Member
4. _____ Dr. Md. Nasim Akhtar Professor Department of Computer Science and Engineering Dhaka University of Engineering & Technology Gazipur–1707, Bangladesh.	Member (External)

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature, and the work of others have been referred to. This thesis or any part of it has not been submitted elsewhere for the award of any degree or diploma.

Author: Dhrubajyoti Das .....
(signature)

Date: November 28, 2023

It is my genuine gratefulness and warmest regard that

I dedicate this work to my beloved

Father and Mother

Table of Contents

List of Figures

List of Tables

List of Terms and Abbreviations

CNN	Convolutional Neural Network
NMS	Non Maxima Suppression
SPPF	Spatial Pyramid Pooling Fast
YOLO	You Only Look Once

Acknowledgement

I am delighted to take this moment to express my profound gratitude towards those whose unwavering support and encouragement were pivotal in the successful completion of my thesis on trash classification using deep neural network.

First and foremost, I would like to convey my heartfelt appreciation to my supervisor, Professor Dr. Kaushik Deb. His unwavering guidance and support were absolutely indispensable throughout this research journey. Without his expert supervision, this report would never have come to fruition. I extend my everlasting thanks to the esteemed members of the examination board for their invaluable insights and suggestions. Your contributions have been instrumental.

I also want to express my gratitude to the entire faculty and staff of the Department of Computer Science and Engineering at CUET. Their support has been a pillar of strength for me.

Special recognition goes to my friends and colleagues for their invaluable guidance and support, which significantly enriched the quality of this research work.

My deepest gratitude and respect are reserved for my parents. Their enduring kindness, sacrifices, and unwavering support have been the bedrock of my journey.

Finally, I humbly acknowledge the divine strength and courage bestowed upon me by Lord Shiva, the Almighty. It is through His grace that I found the perseverance to achieve this significant milestone.

Abstract

Trash production and disposal have emerged as serious issues for underdeveloped nations as their populations have swelled. As manual classification can be both time-consuming and potentially dangerous, therefore, nowadays, it is increasingly being replaced by automated methods. Recent advances in AI and deep learning have allowed for significant advancements in trash detection and classification systems. Due to the lack of a comprehensive trash detection dataset tailored to Bangladesh, we set out to collect data that would accurately portray the complexity of Bangladesh's scenario while also incorporating openlittermap. In this study, we employ a deep learning model known as YOLOv5. Several variants of the YOLOv5 model are used and assessed with both the freshly minted dataset and the already existing benchmark datasets. Simulation results indicate that the finetuned YOLOv5 model outperforms existing models in terms of mean average precision (mAP) and F1-score. On the Bangladeshi dataset, the model shows an mAP of 34.3% and an F1-score of 43.7%. The mAP and F1-score provide a holistic evaluation of YOLOv5's object recognition accuracy, localization, and precision-recall balance. By incorporating the additional data from openlittermap into the new dataset, the mAP is increased to 45.4%. In addition, for some variants of YOLOv5, the suggested model produces greater mAP than the current literature on both the TACO and PlastOpol datasets. The model also achieves an mAP of 84.4% and an F1-score of 78.2% in single-class detection experiments with the newly created dataset. This is because concentrating on just one class helps eliminate class ambiguity, improves localization accuracy, and mitigates class imbalance.

Keywords: Bangladeshi trash, trash dataset, trash detection, transfer learning, single-stage object detector.

Chapter 1

Introduction

1.1 Introduction

In today's world, trash littering is a massive problem that affects many countries. The problem is exacerbated in underdeveloped countries due to inadequate waste-handling infrastructure and resource restrictions. Even industrialized countries, however, are not immune to the problem and confront considerable hurdles in managing their trash. Managing such large amounts of trash is still troublesome for many countries, especially in low-income nations where waste management systems may be insufficient or non-existent [1]. Poor trash disposal may lead to a variety of environmental and health issues. There are not enough waste management facilities in many regions of the world to handle the amount of trash produced. This can result in trash accumulating in public places or being deposited in unapproved areas, producing pollution and health risks. Additionally, many impoverished countries lack the financial and human resources to implement effective waste management systems. Another concern is a lack of public awareness about the importance of proper trash disposal. Many individuals are uninformed of the dangers of improper garbage disposal and may be unsure how to dispose of various types of waste properly.

Different types of trash are found in nature, including organic, inorganic, toxic, electronics, construction, agricultural, and municipal solid trash. According to the World Health Organization (WHO) [2], around 15% of healthcare waste is infectious, poisonous, or radioactive; the remaining 85% is general, as shown in Figure 1.1.

In a developing country like Bangladesh, which has a population of over 160 million and limited resources to deal with waste management, the country has struggled to keep up with the increasing amount of trash produced daily. In terms of solid waste generation, urban areas in Bangladesh generate around 25,000 tons of trash each day, with an annual average of 170 kg per inhabitant [3]. Dhaka, the capital city of Bangladesh, alone generates over 4500 tonnes of trash daily, putting severe pressure on the city's inadequate waste manage-

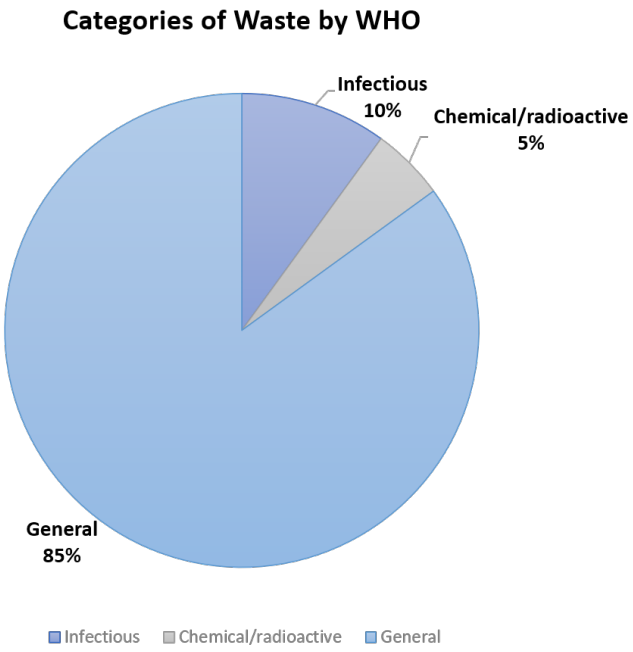

Figure 1.1. Waste categories by WHO

ment infrastructure [4]. The management of solid waste is a major issue for Bangladesh due to the fact that by the year 2025, the quantity of garbage generated per capita will reach 0.75 kilograms per capita per day, and the overall amount of waste will reach 21.07 million tons per year [5]. To keep the area clean and healthy, enormous amounts of trash must be managed. One way to manage trash is to sort it into recyclables, compostables, and non-recyclables [6]. It also reduces the amount of trash put in landfills. Waste management procedures have changed due to the COVID-19 pandemic. Medical waste and single-use plastics have increased due to the increased usage of personal protective equipment (PPE) and disposable items.

Due to the fact that trash does not conform to any certain shape, the process of categorization can be rather challenging. Several scholars have made significant contributions to solving the challenges of waste categorization. Cutting-edge technologies have effectively addressed the challenges of categorizing trash in recent years. Recent studies have aimed to automate the trash classification process by employing cutting-edge technologies like machine learning and computer vision. Recyclables, compostables, and non-recyclables have all been successfully detected and classified using deep learning techniques. Convolutional neural networks (CNNs) have been utilized for feature-based waste item recognition [7]–[10], and transfer learning has made it possible to train models with limited data. Certain tasks are oriented toward the detection of multiple classes, while others are solely focused on the detection of a single class. In this study, we use a single-stage object detector to conduct both types of tests and present a comprehensive analysis of the results. For

this purpose, we opted for a lightweight object detection model. Due to its efficiency and rapidity in image processing, a lightweight object detection model is favored since it enables object identification in real-time and performs well in environments with limited resources. The ability to deploy the model on smaller devices is another bonus of using a lightweight object detection model, which is determined by the model's tiny size and fast inference time [11].

The benchmark datasets that are accessible for these activities, however, are still quite restricted in number. Regarding the classification problem, the popular benchmark dataset currently accessible is called TrashNet [12], consisting of 2527 images, and it has been utilized in various research studies. TACO [13] is a benchmark dataset that is commonly used for the detection task. It consists of 1500 pictures with 4784 annotations. The majority of the public datasets that might be used for classification tasks do not take into account the actual backgrounds. As a consequence of this, the generalization is somewhat inaccurate because the image has a genuine background. In addition, the image that is included in the dataset only shows a single garbage object. However, the background may have several pieces of trash in a single image. In this scenario, the items must be identified before being categorized. There are many different object detection models, such as R-CNN [14], Faster R-CNN [15], Mask R-CNN [16], SSD [17], EfficientDet [18], and YOLO [19], which are popular among researchers for detection tasks. In this research, we endeavored to compile a dataset by taking into account the actual conditions that prevail in Bangladesh. For the purpose of expanding our dataset, we additionally obtained more images from openlittermap [20].

The structure of this chapter is as follows: It commences with a concise introduction to the proposed trash classification system, followed by an enumeration of the challenges faced. Subsequently, the applications of the trash classification system are delineated. The motivation and contributions of this thesis are expounded upon. Lastly, the chapter concludes with an overview of the entire thesis.

1.2 Trash Classification System

Trash classification using deep neural networks is a supervised learning methodology employed in the domain of computer vision with the objective of categorizing diverse forms of waste materials based on visual data, such as photos or videos. In recent years, there has been a notable increase in the attention given to this subject, mostly driven by its relevance to environmental concerns and waste management practices. This includes its use in areas like recycling, pollution mitigation, and the preservation of valuable resources. Within the realm of waste categorization, there are two primary responsibilities that can be

discerned:

1.2.1 Single Trash Classification

The process of single trash classification entails the identification and categorization of individual trash pieces. The aforementioned things encompass a diverse range of waste materials, including but not limited to plastic bottles, paper, glass, and technological garbage. Deep neural networks, namely convolutional neural networks (CNNs), are frequently employed for this particular undertaking. Convolutional Neural Networks (CNNs) have demonstrated efficacy in the extraction of distinctive characteristics from images enabling the identification and classification of different types of waste materials. The efficacy of a unified waste categorization system relies on its ability to accurately process images with varying viewpoints, lighting conditions, and backdrops.

1.2.2 Multi Trash Classification

In contrast, multi-trash classification expands the functionalities of trash classification systems by identifying and classifying many trash pieces concurrently inside a particular scene or frame. The methodology necessitates the first identification of the trash objects inside the picture. In order to achieve this objective, object identification models such as YOLO (You Only Look Once) or Faster R-CNN (Region-based Convolutional Neural Network) can be utilized. Object detection models are specifically engineered to accurately recognize and pinpoint things of interest inside an image. In the context of this particular scenario, the objects of interest pertain to the various kinds of rubbish. After the identification of trash objects, the system proceeds to categorize each of them into their appropriate classifications. In order to optimize the efficacy of multi-trash classification, it is important to effectively tackle the obstacles related to the identification and categorization of various trash objects that possess diverse forms, sizes, and spatial orientations within a singular frame.

The major steps of developing our proposed architecture are summarized here, such as,

- Develop a trash classification dataset that contains both single-trash and multi-trash categories
- Split dataset into the training set, test set, and validation set
- Perform annotation for the dataset
- Preprocessing
- Select classification model

- Training and hyperparameter tunning
- Extract spatial features using CNN
- Generate output probabilities of each trash category

1.3 Challenges

The categorization of trash using deep neural networks presents several obstacles, particularly when used in outdoor environments. The following are few prominent challenges:

- **Identifying multiple trash objects from outdoor scenarios:** Outdoor locations often present visually intricate backgrounds, which can make it difficult for models to effectively distinguish between discarded goods and natural elements like trees, grass, or bodies of water. The existence of various lighting conditions in outdoor settings might bring unpredictability, which in turn can affect the visual quality of photographs and create challenges in accurately identifying trash things. The complexity of identifying and categorizing trash items may increase as a result of their potential to be partially or fully obscured by surrounding objects.
- **Limited range of classes in existing studies:** Numerous extant research and datasets have a narrow scope, concentrating on a restricted number of categories pertaining to waste materials. This phenomenon has the potential to generate models that are highly compatible with certain locations or waste management systems, however, may lack adaptability when confronted with diverse settings including a broader range of trash categories. As an example, The categorization of medical waste, which presents distinct health and safety hazards, necessitates the utilization of specialized models and training data. The identification and management of medical waste within the framework of waste categorization provide a significant and pressing obstacle.
- **Data imbalance:** Imbalanced datasets, characterized by the underrepresentation or overrepresentation of specific trash classes, have the potential to introduce bias into the performance of models. The rectification of data imbalance is of utmost importance in order to attain precise and equitable outcomes in waste categorization.
- **Lack of comprehensive datasets from the Bangladeshi environment:** The presence of diverse and high-quality datasets plays a pivotal role in the training and evaluation of waste categorization algorithms. The absence of comprehensive statistics pertaining to the local environment, waste kinds, and disposal practices in Bangladesh might impede the formulation of efficacious remedies. The collection and curation of datasets that contain a comprehensive array of waste materials often encountered in Bangladesh, including culturally distinctive things, are crucial for the development of models that can successfully operate within this particular

environment.

1.4 Applications of Trash Classification System

- **Construction of intelligent smart waste sorter:** Trash categorization systems play a crucial role in the advancement of smart garbage sorters, which find application in diverse waste management and recycling facilities. These systems employ computer vision and deep learning algorithms to autonomously categorize and separate many sorts of waste products, including plastics, glass, paper, and electronic waste. The primary advantages encompass enhanced efficiency in recycling processes, reduced levels of contamination in recyclable materials, and heightened levels of resource recovery. The use of intelligent trash sorting systems plays a significant role in promoting sustainable waste management strategies, hence yielding both economic and environmental advantages.
- **Monitoring of illegal dumping:** Trash categorization systems have the potential to be implemented in outdoor settings and public spaces for the purpose of monitoring and identifying instances of illicit waste disposal. The utilization of surveillance cameras integrated with computer vision technology enables the detection of situations in which persons or organizations engage in the unauthorized disposal of waste in locations not approved for such purposes. The system has the capability to initiate warnings or notifications to law enforcement or environmental authorities, prompting them to undertake necessary measures against individuals engaged in unlawful dumping activities. This application serves the purpose of deterring and mitigating unauthorized rubbish dumping, so safeguarding the cleanliness and visual appeal of public places while also ensuring environmental protection.
- **Waste management and recycling:** One of the principal uses pertains to waste management facilities and recycling centers. Trash classification systems have the capability to automate the process of classifying recyclable materials, such as plastics, glass, and paper, from non-recyclable items. This automation significantly enhances the efficiency and precision of the recycling process.

1.5 Motivation

The motivation behind the implementation of a waste categorization system stems from the urgent necessity to mitigate the limitations and risks inherent in manual classification. Manual classification is susceptible to misidentification and human fallibility, thereby necessitating a more reliable alternative. Furthermore, the absence of proper identification for workers handling hazardous waste exposes them to considerable health hazards, which

must be effectively addressed. In addition, the absence of accessible datasets pertaining to the Bangladeshi context underscores the need for customized approaches that can address the unique requirements of waste management in the local setting. These tailored solutions are essential in surpassing the constraints of previous research endeavors, thus enabling more efficient and accurate waste management practices, safeguarding the environment, and ensuring the well-being of workers.

1.6 Contribution of the Thesis

- Construct a diversified collection of 4418 images, incorporating data from Bangladesh and openlitremap. There are ten distinct types of trash in the dataset. The Bangladeshi dataset has 1283 images, while openlittermap provides 3135 images. Due to the scarcity of benchmark datasets for the detection work, our dataset will be able to fulfill the requirement of the challenge, which is to concentrate data on the Bangladeshi environment.
- Apply a transfer-learning-based object detection model that has been adequately fine-tuned for trash detection tasks using the COCO dataset and then conduct experiments utilizing the selected fine-tuned models on our datasets. The experimental outcomes validate the practicability of the approach. Analyze and compare our models' accuracy to that of state-of-the-art models using two publicly available datasets. According to the findings of the analysis, the accuracy of our models was superior to that of the most recent and cutting-edge model.

1.7 Thesis Organization

The subsequent sections of this thesis are structured as follows. Chapter 2 provides an overview of pertinent technology. In this section, a concise overview of the research pertaining to other connected studies is provided. Chapter 3 focuses on the suggested architectures, encompassing comprehensive graphics that are essential for understanding the subject matter. In Chapter 4, a comprehensive overview is provided of the experimental outcomes pertaining to the suggested structures, encompassing an assortment of accuracy metrics. The last chapter serves as the concluding section of the entire body of work. Additionally, this chapter also references several forthcoming research endeavors.

1.8 Conclusion

This chapter presents a comprehensive review of the notion of trash classification, encompassing the inherent obstacles associated with its implementation. Furthermore, an

extensive examination was conducted on the diverse range of applications pertaining to the classification of trash, with particular emphasis placed on elucidating its pertinence and significance. Subsequently, we expounded upon the underlying rationale and the significant contributions of this study endeavor. The forthcoming chapter will undertake a comprehensive examination of the pertinent literature, providing more context to the subject area.

Chapter 2

Literature Review

2.1 Introduction

To enhance comprehension of contemporary techniques within the domain of trash classification systems, we present a succinct outline. This comprehensive review examines the core ideas in deep learning, focusing on single-stage and multi-stage object detectors, as well as several classification models for single trash items. The key principles encompassed in this discussion are Object detectors, feature pyramid networks (FPN), transfer learning, Faster R-CNN, and YOLOv5. Furthermore, the activation functions and categorical cross-entropy loss functions that are frequently utilized in these models are also addressed. Following this, we proceed to examine comparable studies in the domain of trash classification, organizing them according to whether they concentrate on distinct waste items within the same waste category or contain a variety of categories of trash items. The chapter is structured in the following manner: At first, a concise overview of single-stage and multi-stage object detectors is presented. Subsequently, the process of transfer learning and its use in the domain of trash classification are introduced. In the end, we present a thorough examination of current methodologies, with succinct explanations of each technique and their significance within the realm of trash categorization systems.

2.2 Object Detectors

The task of object detection in computer vision pertains to the identification and localization of objects inside an image. The procedure of object detection generally encompasses the subsequent stages:

(i) **Input Image:** The initial step involves the utilization of an input image, which has the potential to encompass many things that are of significance.

(ii) **Feature Extraction:** Feature extraction involves the processing of an image to identify and extract pertinent features. This stage entails the utilization of convolutional

neural networks (CNNs) to effectively capture and analyze patterns and distinctive features present in the image.

(iii) **Candidate Region Proposal:** The utilization of region suggestion methods by object detectors is employed to effectively limit the search space and locate prospective object locations. This procedure involves the creation of bounding boxes that enclose regions inside the image that are likely to contain objects.

(iv) **Classification:** The regions that have been suggested are subjected to more in-depth analysis in order to determine whether they possess an object and, if they do, to ascertain the specific sort of object it represents. This process entails the utilization of classification models to allocate labels to individual regions.

(v) **Bounding Box Refinement:** The process of bounding box refinement involves the adjustment of bounding boxes in order to more correctly encompass the items that have been detected. Frequently, this entails modifying the placement and dimensions of the bounding boxes.

(vi) **Object Localization:** Object localization is the concluding stage of the process, wherein the coordinates of the object within the image are provided, facilitating accurate and precise object localization.

There are two main methodologies for object detection, namely single-stage and multi-stage object detectors.

(i) **Single-Stage Object Detectors:** Single-stage object detectors are designed to predict the bounding boxes and class labels of all objects in a single run through the network without the need for several stages or iterations. Typically, these types exhibit higher speed but may exhibit somewhat reduced accuracy in comparison to multi-stage detectors. Two well-known examples of single-stage detectors are YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector).

(ii) **Multi-Stage Object Detectors:** Multi-stage object detectors adhere to a dual-step procedure. Initially, prospective item placements are identified, commonly known as region suggestions. Subsequently, these zones undergo classification and refinement. Multi-stage detectors have been observed to exhibit greater accuracy rates but at the expense of increased processing requirements. One widely recognized instance is the Faster R-CNN, which stands for Region-based Convolutional Neural Network.

In brief, the process of object detection entails the recognition and precise localization of items inside an image. Single-stage detectors strive to accomplish this objective in a solitary iteration, whereas multi-stage detectors adhere to a more intricate, bipartite procedure. The selection of these methodologies is contingent upon the precise demands of the undertaking while also considering the trade-off between expeditiousness and precision.

2.3 Single-Stage Object Detectors

The objective of single-stage object detection is to identify and categorize things inside an image by utilizing a single iteration through the neural network. The system is renowned for its high level of efficiency and rapidity. The procedure generally encompasses the subsequent stages:

(i) **Input Image:** The process commences with an input image.

(ii) **Feature Extraction:** The process of feature extraction involves the utilization of a convolutional neural network (CNN) to analyse the image and extract relevant information. This stage involves the identification of patterns, edges, and pertinent information contained within the image.

(iii) **Grid-Based Prediction:** Grid-based prediction involves dividing an image into a grid of cells, wherein each cell is assigned the task of predicting a collection of bounding boxes by the model. The bounding boxes encompass pertinent data pertaining to their coordinates (x, y, width, height), confidence score (indicating the likelihood of containing an item), and class scores (representing the scores assigned to each conceivable object class).

(iv) **Non-Maximum Suppression (NMS):** Non-Maximum Suppression is employed as a means to eliminate redundant and less reliable forecasts. The process involves eliminating redundant bounding boxes by selecting the one with the highest confidence score.

(v) **Class Labeling:** Class labeling involves the assignment of class labels to the bounding boxes. The labels assigned by the model are determined by selecting the class score with the highest level of confidence for each individual box.

(vi) **Output:** The ultimate result comprises the coordinates, class labels, and confidence ratings pertaining to the identified items. The findings are displayed in the form of bounding boxes encompassing the identified objects.

2.3.1 Examples of Single-stage object detectors

Examples of two single-stage object detectors are given below:

YOLO: The YOLO system is widely recognized in the field of computer vision for its ability to perform real-time object recognition using a single-stage approach [19]. The developmental trajectory of YOLO has made a substantial contribution to the advancement of object identification techniques. The purpose of its design was to bring about a revolutionary change in the process of detecting and categorizing objects within photos

and videos, accomplishing this task in a singular, efficient iteration through the neural network. One of YOLO's fundamental innovations lies in its grid-based approach. The input image is partitioned into a grid of cells, wherein each cell has the task of predicting multiple bounding boxes. The bounding boxes contain crucial data such as coordinates, confidence scores that measure the probability of object presence, and class probabilities that determine the classifications of the objects. An additional crucial element in the development of YOLO is the integration of non-maximum suppression. This technique effectively eliminates bounding boxes with low confidence levels and redundancy, ensuring that only the most precise predictions are preserved. The concept of YOLO has undergone continuous development, resulting in its significant recognition for its ability to deliver real-time performance. The speedy and precise object detection capabilities of this technology make it a highly appealing option for applications that require prompt responsiveness. Moreover, the Darknet architecture is intricately linked with YOLO, a neural network framework that is specifically designed to enhance and align with the aims of YOLO. The significant incorporation of this close integration has played a crucial role in the historical development of YOLO, hence contributing to its extensive acceptance and achievements within the realm of computer vision and object identification.

SSD (Single Shot MultiBox Detector): The Single Shot MultiBox Detector (SSD) has made a substantial impact on the advancement of object identification and possesses a distinctive background within the domain. The development of SSD aimed to strike a harmonious equilibrium between precision and efficiency, rendering it a versatile and broadly usable tool across diverse domains. One significant advancement in the field of object detection is the ability of SSD (Single Shot MultiBox Detector) to effectively recognize objects with varying scales and aspect ratios. In order to achieve this objective, SSD utilizes a sequence of convolutional layers that possess varied receptive field sizes [17]. This enables the model to make predictions regarding bounding boxes at many scales, hence effectively capturing objects of diverse sizes present inside images. Another characteristic of SSD is its ability to anticipate bounding boxes with varying aspect ratios, a crucial attribute for accepting objects with diverse geometries. This innovation enhances its efficacy across a diverse array of item categories. The SSD model is capable of predicting class scores for several object categories, enabling it to classify things into a wide range of predefined classes. This characteristic makes it a versatile and well-suited choice for applications that require diverse labeling requirements. Similar to the concept of "You Only Live Once" (YOLO), Single Shot MultiBox Detector (SSD) integrates non-maximum suppression (NMS) as an essential element inside its item identification methodology. This methodology guarantees the elimination of bounding boxes with low confidence and redundancy, resulting in the retention of just the most dependable and precise detections. The SSD model has seen tremendous development and has emerged as a prominent model

in the field of object identification. It provides a favorable trade-off between accuracy and speed, making it highly suitable for a wide range of applications. Moreover, SSD has made substantial contributions to the continuous advancements in the field of computer vision.

2.4 Multi-Stage Object Detectors

The process of multi-stage object detection entails a two-fold methodology, which initially entails identifying probable locations of objects and subsequently involves the classification and fine-tuning of these identified places. The aforementioned approach frequently yields a greater degree of precision, albeit at the cost of increased computing complexity when contrasted with single-stage detectors. The process can be delineated as follows:

(i) **Region Proposal:** The input image undergoes an initial processing step to provide a collection of region proposals. The aforementioned recommendations refer to specific locations inside the image that are very probable to encompass items. This phase is centered on the identification of areas of interest.

(ii) **Feature Extraction:** Feature extraction involves utilizing region proposals to extract relevant features. These qualities aid in the characterization of the regions and facilitate their preparation for subsequent study. Convolutional neural networks (CNNs) are frequently utilized for this particular undertaking.

(iii) **Classification and Localization:** In the subsequent phase, the attributes derived from the region recommendations are employed for the purposes of object categorization and location. The process involves the categorization of objects into predetermined classifications, followed by the determination of their exact positions within the respective regions.

(iv) **Non-Maximum Suppression (NMS):** Non-Maximum Suppression is a commonly employed technique to enhance the quality of results. It involves the elimination of redundant and low-confidence detections, thereby retaining just the most precise predictions.

2.4.1 Examples of Multi-stage object detectors

Examples of two multi-stage object detectors are given below:

R-CNN: The Region-based Convolutional Neural Network (R-CNN) holds a significant position in the historical progression of object detection [14]. The advent of this technology represented a notable advancement in the progression of object-detecting methodologies. The historical development of R-CNN may be traced back to a sequence of important

advancements. The introduction of R-CNN marked a significant advancement in computer vision with the introduction of selective search, a region proposal technique that played a crucial role in the identification of possible object-containing regions inside an image. This innovative methodology established the foundation for object detection based on regions. Selective search is an integral element of the R-CNN framework, as it systematically generates region recommendations. The aforementioned ideas functioned as the initial selection of regions of interest within the input image. Another notable characteristic of R-CNN was its incorporation of pre-trained convolutional neural networks (CNNs), such as AlexNet or VGG, for the purpose of extracting features from the area suggestions that were created. The collected features were important in characterizing the zones for future object detection. The R-CNN model has exhibited a high level of competence in both object classification and localization tasks. The retrieved attributes were utilized to classify objects into predetermined categories and accurately ascertain their spatial positions inside the regions. In order to improve the precision of its detections, R-CNN implemented the technique of non-maximum suppression (NMS). The utilization of this post-processing technique played a crucial role in the filtration of predictions with low confidence and redundancy, hence guaranteeing the selection of the most reliable outcomes. Although R-CNN introduced innovative ideas to the domain of object detection, it encountered computational efficiency issues, particularly with regard to the speed of inference. The continuing value of this model rests in its pioneering role, which established the groundwork for succeeding models to enhance and progress object detection. As a result, both the speed and accuracy of object detection have been improved.

Faster R-CNN: The Faster R-CNN model is a significant milestone in the advancement of object identification, building upon its predecessor, R-CNN. The fundamental objective of this evolutionary approach was to address the speed and efficiency issues encountered by previous methodologies. The introduction of Faster R-CNN brought about a significant innovation in the field, namely the incorporation of the Region Proposal Network (RPN) as a fundamental component within the model. The Region Proposal Network (RPN) was designed with the purpose of generating region proposals directly from convolutional feature maps, hence removing the need for external region proposal algorithms [15]. This innovation has significantly enhanced the efficiency of the object-detecting procedure. The concept of feature sharing was developed by Faster R-CNN to enhance efficiency. This approach enables the model to share convolutional features across the Region Proposal Network (RPN) and the succeeding components responsible for object categorization and localization. The implementation of this upgrade has greatly improved computational efficiency and has made a major contribution to the overall performance of the model. One of the key advancements of Faster R-CNN was the integration of the Region Proposal Network (RPN) with the object detection network, resulting in a unified architecture. The

process of simplifying the structure had a significant impact on improving the speed and efficiency of the model. Similar to its predecessor, R-CNN, Faster R-CNN maintains the use of non-maximum suppression (NMS) as an essential post-processing procedure. The National Monitoring System (NMS) consistently prioritized the retention of highly accurate and reliable detections, so efficiently removing forecasts with low confidence levels and redundant information. The historical account of Faster R-CNN highlights its notable influence on the field of object identification, presenting a revolutionary approach that greatly enhanced the speed and effectiveness of this procedure. The aforementioned model has since emerged as a fundamental concept in the discipline, establishing the foundation for following advancements in the development of expeditious and effective approaches for object detection.

2.5 Related Review for Trash Classification System

This section provides a summary of previous studies on trash classification and detection. Since the development of Deep Learning frameworks, researchers have been employing a wide range of deep learning models to solve their problems. ResNet, MobileNet, DenseNet, and EfficientNet are just a few examples of well-liked classification models. Single-stage object detectors and two-stage object detectors are two methods for accomplishing the detection task. Single-stage object detection method employs a single neural network to do both object localization and classification. These models are often easy to implement and take less time to execute; however, they may be less accurate than their two-stage counterparts. YOLO and SSD are two examples of single-stage methods for object detection. Two-stage object detection divides the detection process into two phases: creating object proposals and then classifying those proposals. The first phase creates a collection of bounding boxes (object proposals) that probably contain objects, while the second phase classifies and refines the locations of these proposals. One-stage models are often faster, while two-stage models are typically more accurate. Faster R-CNN and Mask R-CNN are two-stage object detectors. The following is a synopsis of several relevant works.

2.5.1 Trash Classification

There are few works that solely concentrate on the classification task. A classification task involves sorting an image's garbage into several predetermined categories. The following provides descriptions of some of the more significant works.

- A deep learning-based EfficientNet Architecture was suggested by Masand et al. [21]

to categorize the various types of garbage with greater accuracy and a reduced number of parameters. They combined four different datasets, including TrashNet, Openrecycle, TACO, and Waste Classification, to produce a new dataset with 8135 records. In addition to this, they improved the use of adaptive gradient clipping in order to optimize the models in regions with higher loss. With EfficientNetB3, they obtained an accuracy of 91.87% on their suggested dataset, whereas on the TrashNet dataset, they reached 98%.

- Yang et al. [22] integrates garbage classification and recognition methods of image classification and object detection. They used ResNet and MobileNet for training and testing the dataset. They also used YOLOv5 models for the detection of trash data. They improved the classification accuracy over 2% by integrating the consensus voting algorithm (CVA). They obtained an accuracy of 98%, while without CVA, it was 95%.
- Deep learning and an embedded Linux system were utilized by FU et al. [23] in the development of their intelligent trash categorization systems. They used a Raspberry Pi 4B for the hardware implementation of the project. In addition, for the software implementation, they utilized GNet, which is a combination of transfer learning and an upgraded version of the MobileNetv3 model, was the algorithm that they utilized for the classification test. Finally, they constructed a human-computer interaction system in order to carry out efficient monitoring of the system. They carried out their trials using the dataset provided by the Huawei Trash Classification Challenge Cup, and their results showed an accuracy of 92.62%.
- For better results, Vo et al. [24] suggested a method using the DNN-TC model, an improved version of the ResNext model. They sorted 5,904 photos of trash found in Vietnam (VN Trash) into three groups: organic, inorganic, and medicinal. Both the VN Trash dataset and the TrashNet dataset serve as benchmarks for trash classification, and their results were compared to those of DNN-TC and other state-of-the-art algorithms. The results indicated that DNN-TC has an accuracy of 94% for the TrashNet dataset and 98% for the VN Trash dataset.

2.5.2 Trash Detection

If an image contains numerous types of items, detection is required. Researchers have used both one-stage and two-stage object detection approaches to look for trash. Here we provide summaries of a few representative studies.

- Cordova et al. [25] compared various state-of-the-art object identification techniques, such as Faster R-CNN, Mask R-CNN, Efficient- Det, RetinaNet, and YOLOv5 for trash detection. Moreover, they created a whole new dataset of 2418 images called

Plastopol. Furthermore, they also employed TACO, a standard waste dataset in addition to Plastopol. For both datasets, they found that YOLOv5 models performed better than other models. They focused on only one category in both datasets and got a mAP of 84.9% on Plastopol and 63.3% on the TACO dataset, respectively.

- A deep learning-based i-YOLOX model for trash identification was proposed by Liu et al. [26]. A new dataset was developed that was inspired by the actual world. They added involution, a simpler structure than convolution, to the original model. In addition, a convolution block attention module (CBAM) was added to the original model to enhance the feature extraction process. They observed that their proposed model outperformed the existing state-of-the-art models. i-YOLOX showed an improvement by 1.47% and cut down on parameters by 23.3% in comparison with existing models.
- Improved YOLOv5s model for finding garbage in the ocean was proposed by Wu et al. [27]. In order to improve the design of the original YOLOv5s, they swapped out the network infrastructure with a MobileNetv3 system. To enhance the feature extraction capabilities, they also used CBAM. They utilized the underwater garbage images from ICRA19-Trash. The dataset was randomly divided into 80, 10, and 10 halves. With the ICRA19-Trash dataset, they reached an accuracy of 97.5% on their test dataset. The detection speed, however, is 2.5 times slower than the standard YOLOv5s model.
- To find trash in the ocean, Tian et al. [28] proposed a modified version of the YOLOv4 model. Based on the original YOLOv4 model, they developed a new four-level detection strategy. In addition, they used model pruning to condense their model by getting rid of unnecessary weights. They managed to get a 95% mAP.
- Experiments using a variety of object detection techniques were carried out by Melinte et al. [29] on the TrashNet dataset. The TrashNet dataset typically only takes into consideration a single trash item in an indoor setting. They reached the highest level of accuracy possible with SSD, which is 97.63%. They were able to attain an mAP of 95.76% by utilizing the Faster R-CNN algorithm.

Table 2.1 summarizes the relevant literature based on the problem category, model utilized, dataset availability, and accuracy.

Table 2.1. Summary of detection-based related works

Reference	Model used	Dataset availability	mAP
Yang et al. [22]	YOLOv5	No	98%
Cordova et al. [25]	YOLOv5	Plastopol : Yes, TACO : Yes	Plastopol: 84.9%, TACO: 63.3%
Liu et al. [26]	iYOLOX	No	97.57%
Wu et al. [27]	YOLOv5	ICRA19 : Yes	97.5%
Tian et al. [28]	YOLOv4	No	95%
Melinte et al. [29]	SSD	TrashNet : Yes	97.63%

2.6 Conclusion

This chapter is devoted to a comprehensive analysis of the extant literature. The text offers an extensive examination of diverse deep learning algorithms and investigates alternative approaches relevant to the classification of waste materials. The following chapter will explore the process of the proposed trash classification system.

Chapter 3

Proposed Architecture

3.1 Introduction

The process of classifying waste in real environmental images has significant intricacies. The obstacles involved in this context include factors such as unanticipated variations in lighting conditions, diverse image views, obstructions caused by objects, and the inherent similarity between different categories of waste.

Nevertheless, the present study has effectively addressed these challenges and established a pioneering dataset encompassing a diverse range of waste categories and several subtypes within these categories. The dataset was meticulously curated to incorporate complex environmental variables, such as variations in illumination and a significant level of image similarity. The studies were carried out using RGB images with dimensions of 640×640. In order to effectively represent the spatial characteristics depicted in the environmental photographs, a variety of tailored Convolutional Neural Network (CNN) structures were created within the YOLOv5 framework. These structures consist of the backbone, neck, and head components. The procedure involved the application of transfer learning methods, utilizing the CSP Darknet53 model to enhance the extraction of features. Finally, a softmax layer was smoothly integrated into the YOLOv5 model to produce class predictions, with special care taken to guarantee that the cumulative prediction values equated to 1.

The section's progression encompasses seven crucial stages: (a) the generation of the dataset, (b) the annotation of the dataset, (c) the preprocessing of the dataset, (d) the design of the proposed network, (e) the training process of the network, and (f) the evaluation of the network. The major steps are highlighted in Figure 3.1.

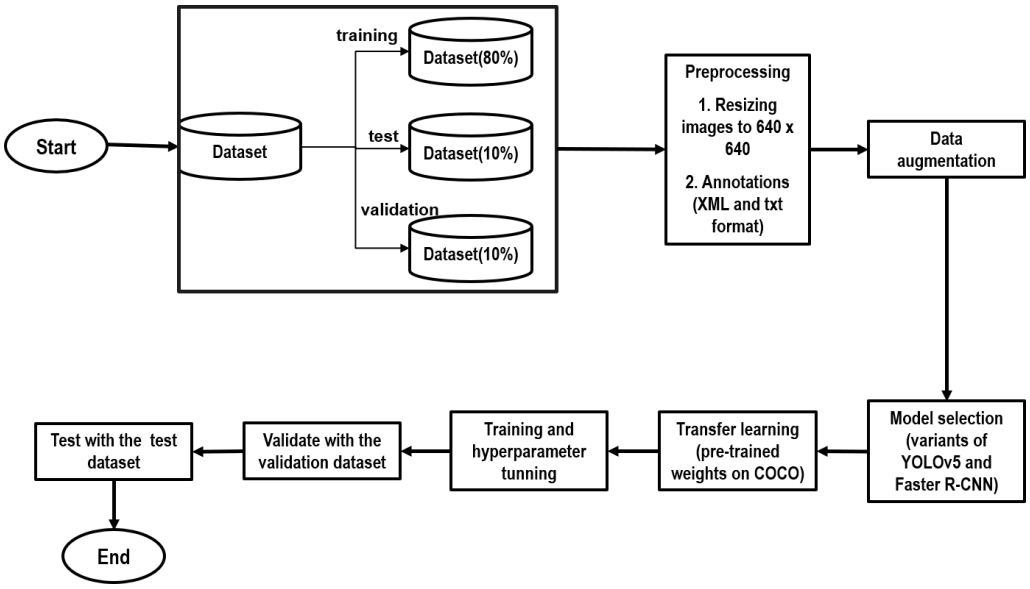

Figure 3.1. Workflow of the proposed trash classification system

3.2 Dataset Generation

The data collection process consists of two parts. Each image in the collection has one or more pieces of trash. Overall, we gathered 1283 images of trash from the nature of Bangladesh. As a whole, our image collection spans over ten distinct classes. Diverse geographical locations in Bangladesh were selected, including roadsides, parks, sea beaches, and some selected village areas. For variety, we have selected images from a wide range of periods. Both single-trash and multi-trash images are included in our dataset.

In contrast to datasets that only take into account static backdrops, we want to create one that more closely resembles the real world. Plastic has the highest number of occurrences out of all the materials in our collection. It also suggests the ever-increasing plastic trash along our streets and other public spaces. Samsung Galaxy M21 was utilized as a photographic device. The camera has a whopping 48 megapixels of resolution. There were a total of 1283 images annotated, and these images contained a total of 6178 trash objects. This information is presented in Table 3.1. We have employed makesense.ai for annotation purposes. Makesense.ai is an online image labeling tool. We utilized ten distinct labels to annotate the images of trash. Next, we downloaded 3135 images from openlittremap. Openlitremap is a database that contains litter and plastic images from all over the world. After that, we added metadata to the openlittremap images we gathered. A total of 3659 trash objects were annotated from this dataset, as shown in Table 3.2. Overall, 4418 images with 9837 trash objects have been manually annotated. Figure 3.2 displays how our datasets are broken up. Several excerpts from our data collection are displayed in Figure 3.3 and

Figure 3.4. Data collection occurred under various lighting conditions, including morning, noon, and afternoon. This variability aimed to capture complex patterns and enhance the dataset's robustness. The intricacies associated with the task of object detection give rise to unique obstacles in achieving data balancing. In contrast to classification, object detection entails the annotation of many objects belonging to various classes inside a single image. The complexity of this situation presents challenges in attaining data balance. In the course of our investigation, we acknowledge these obstacles and have implemented measures to tackle them. The dataset was carefully curated, incorporating images from openlittermap in order to increase diversity and mitigate potential asymmetries.

We used the available datasets TACO and PlastOpol to test how well our chosen models generalize to other datasets. TACO is an open picture collection that documents garbage in its natural environment. It features images of trash taken in a variety of settings, ranging from beachfront locations in the tropics to urban areas in the UK. TACO consists of 1500 images with 4784 annotations supplied in the COCO format, whereas PlastOpol comprises 2418 images with 5300 annotations, as shown in Table 3.3. In Table 3.4, we also presented an overview of the maximum, average, and minimum number of cases in the datasets we used for our research. Our Bangladeshi dataset has a larger mean instance compared to the others. Some samples of existing datasets are shown in Figure 3.5 and Figure 3.6

Table 3.1. Distribution of Bangladeshi dataset

Total image	**Category**	**Instances**
1283	Tissue paper	471
	Plastic	2481
	Medical waste	191
	Rope	112
	Paper	1339
	Cigarette butt and box	1088
	Metal	9
	Glass	1
	Organic waste	336
	Textiles	150
Total		6178

Table 3.2. Distribution of Openlittermap dataset

Total images	Category	Instances
3135	Tissue paper	82
	Plastic	1822
	Medical waste	233
	Rope	85
	Paper	456
	Cigarette butt and box	338
	Metal	526
	Glass	89
	Organic waste	23
	Textiles	5
Total		3659

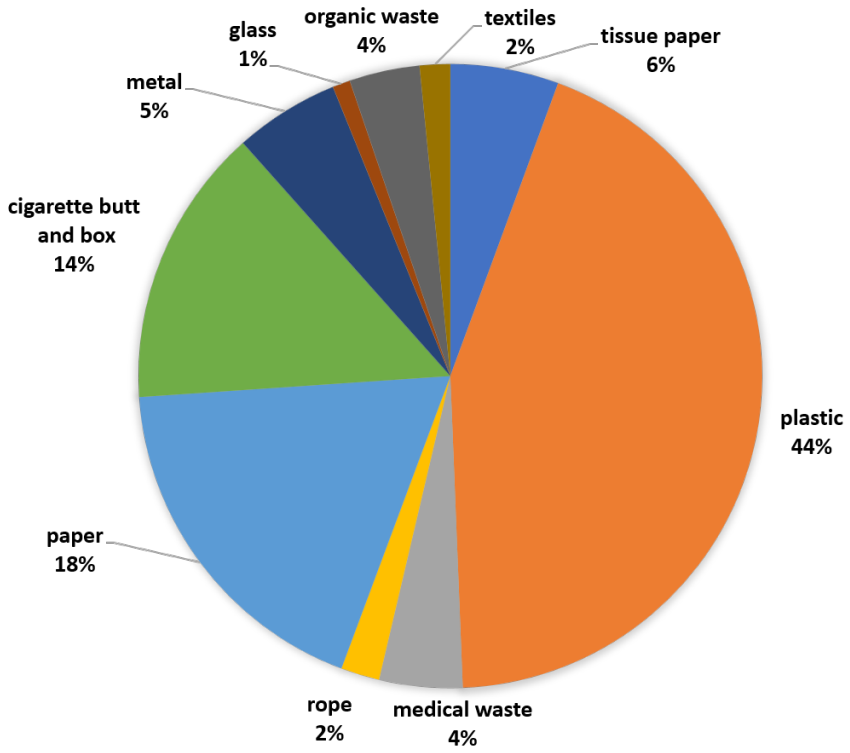

Figure 3.2. Our extended dataset distribution of 10 categories

Figure 3.3. Examples from our Bangladeshi dataset: (a) Trash in daylight, (b) Trash in direct sunlight, (c) Trash in a grassy environment, (d) Trash in a combination of light and shadow, (e) Trash in cloudy conditions, (f) Trash captured in rainy weather

Figure 3.4. Examples from Openlittermap dataset: (a) glass, (b) plastic and metal, (c) cigarette butt, (d) organic waste, (e) medical waste and organic waste, (f) metal

Table 3.3. Summary of existing datasets

Dataset	Total images	Instances	Classes
TACO	1500	4784	60
PlastOpol	2418	5300	Not known

Table 3.4. Summary of the maximum, average, and minimum number of instances in different datasets

Datasets	Max	Average	Min
Bangladeshi dataset	24	5	1
OpenlittreMap	27	2	1
TACO	90	3	1
PlastOpol	40	2	1

Figure 3.5. Samples of PlastOpol dataset

Figure 3.6. Samples of TACO dataset

3.3 Data Preprocessing

Preprocessing plays a crucial role in the process of trash classification utilizing a deep neural network. This stage is crucial for ensuring that the input data is appropriately formatted and of sufficient quality to facilitate the training and testing of the model. The following section outlines the steps involved in data preprocessing.

3.3.1 Resizing

In order to ensure uniformity in the proportions of photos, it is recommended that all images be downsized to a resolution of 640x640 pixels. The utilization of a standard size enables the optimization of processing and training of models, resulting in enhanced efficiency.

3.3.2 Data Augmentation

Data augmentation is a method for expanding a training dataset by generating novel variants of existing images [30]. Data augmentation aims to model contingencies and variations that may arise in the actual world. The more examples of the same item the model sees, the more resilient and generalized it becomes. Data augmentation is a crucial aspect of YOLOv5's training process and may be accomplished in a number of ways. YOLOv5 uses Albumentations [31], a Python module that enables quick and versatile image augmentation. The description of several augmentation techniques is as follows:

- Hue shift augmentation (hsv_h) involves arbitrarily rescaling an image's hue values within a given tolerance.
- Saturation shift augmentation (hsv_s) involves making arbitrary changes to an image's saturation within a predetermined window.
- Value shift augmentation (hsv_v), a method for arbitrarily adjusting an image's brightness within a certain parameter.
- Translation augmentation (translate) Randomly translates a visual object within a certain range.
- Scaling augmentation (scale), wherein the item in the image is scaled arbitrarily within a certain range.
- Flip left-right augmentation (fliplr) flips the image horizontally.
- Mosaic augmentation (mosaic), a method for creating a new training sample by combining four images into one.
- Mixup augmentation (mixup), which combines two unrelated images and their labels to produce a new data set for training.
- Anchors (anchors) are pre-defined boxes of varying sizes and aspect ratios used to estimate the location of objects in an image.

The augmentation strategies that we employed are displayed in Table 3.5.

Table 3.5. Different augmentation techniques used in our selected model

Augmentation techniques	Value
hsv_h	0.01041
hsv_s	0.54703
hsv_v	0.27739
translate	0.04591
scale	0.75544
fliplr	0.5
mosaic	0.85834
mixup	0.04266
anchors	3.412

3.4 Data Annotation

The process of manually annotating the dataset plays a crucial role in the context of supervised learning. In the context of waste categorization, it is imperative to accurately classify the images. The makesense.ai platform was utilized to achieve efficient and precise data labeling. These programs often allow users to create bounding boxes around waste objects seen in the images. The process involves manually delineating each individual trash item seen in the images by employing bounding boxes and utilizing an annotation tool. Following that, we proceed to assign the suitable category label to every enclosing box. It is imperative to acknowledge that although this particular phase may require a significant amount of time, it is crucial in order to effectively train a resilient classification model. The annotations should be stored in a meticulously organized format, such as XML or TXT. This style guarantees a smooth integration with our selected deep learning framework, facilitating the comprehension and usage of the annotated data by the model in a straightforward manner. By adhering to these procedures, we generate a thoroughly annotated dataset that functions as the fundamental basis for training a reliable trash categorization model within the framework of supervised learning.

3.5 YOLOv5 Model

We have chosen YOLOv5 model [32] for our detection task as it provides high speed as well as good accuracy, which is useful in real-time smart applications. We opted for the YOLOv5 model since it is both more accurate and faster than the predecessors, the YOLOv3 [33] and YOLOv4 [34] models. YOLOv5's performance has been enhanced by a number of changes to its architecture, such as the addition of a cross-stage partial backbone, feature aggregation via spatial pyramid pooling, and the adoption of the SiLU activation function. In addition, the YOLOv5 model is lightweight, making it useful in

many application areas [35], [36]. We have experimented with several YOLOv5 models to find the better one for our task. We have used YOLOv5s(small), YOLOv5m(medium), YOLOv5l(large) and YOLOv5x(Extra large). Here 's', 'm', 'l', and 'x' refers to the size of the YOLOv5 model, with s being the smallest and x being the largest. The size of the model is calculated by the number of layers, filters, and channels in the convolution layer. YOLOv5s is the fastest and smallest version of the YOLOv5 family. It has less number of parameters and also takes less time to train. However, the accuracy is a bit of a concern here than in the other versions of YOLOv5. YOLOv5x is the largest version and provides higher accuracy than the others, but it trades speed and training time for accuracy. Also, it has a longer inference time. The YOLOv5 model is composed of three parts, namely the backbone, neck, and head.

3.5.1 Backbone

The backbone of a model extracts useful features which are helpful for object detection tasks. The backbone is made up of a number of convolutional layers that operate on the input image and generate a hierarchy of abstracted feature maps. We used the CSPDarknet53 network, a tweaked take on the Darknet foundation utilized in earlier YOLO versions, which is the basis of the YOLOv5 architecture. Since YOLO relies on residual and dense blocks to transmit data to the deepest levels, it struggles with the redundant gradient. By splitting the feature maps acquired from the input layer in half and then merging the halves through a cross-stage hierarchy, Cross Stage Partial (CSP) networks mitigate this issue [37]. The backbone of YOLOv5 is given in Figure 3.7

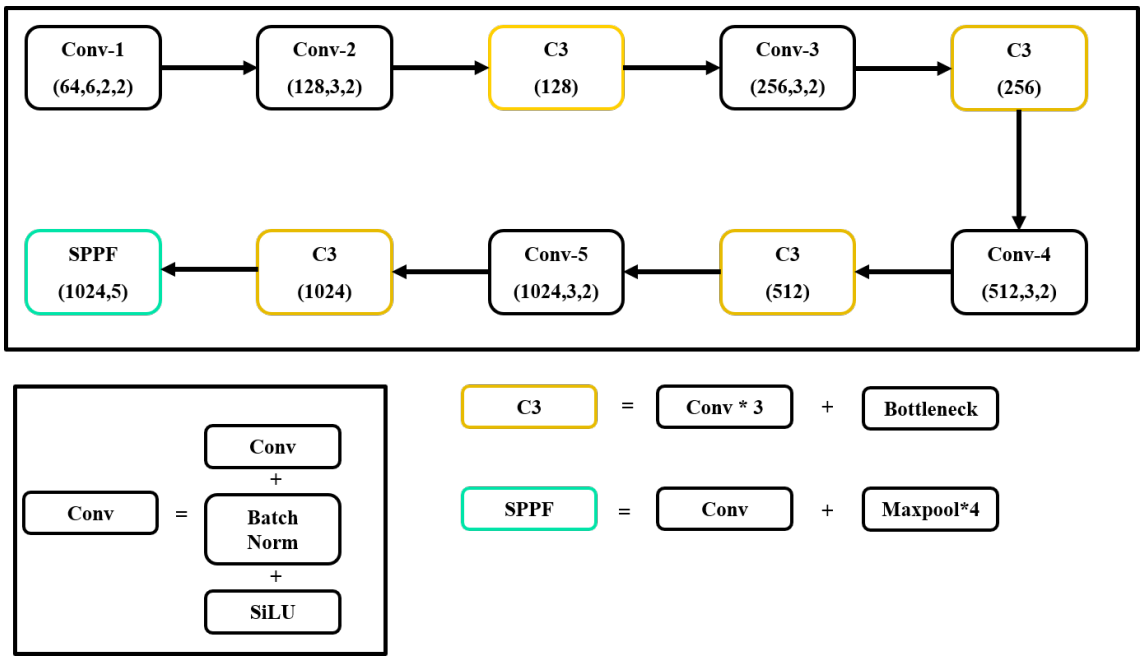

Figure 3.7. Backbone of YOLOv5 model

3.5.2 Neck

The neck in the YOLOv5 architecture is a set of layers that connect the network's backbone to the detection head. The neck serves to combine the high and low-resolution features from the backbone and to extract more discriminative features that are useful for object detection. The spatial resolution of the feature maps is lowered by the neck, which lessens the burden on the network's computational resources. YOLOv5 neck relies on a combination of Spatial pyramid pooling - fast (SPPF) and a path aggregation network(PANet). Using a combined bottom-up and top-down strategy, PANet compiles features across the various layers of the underlying network [38]. SPPF was created to aid the network in dealing with objects of varying sizes and shapes.

3.5.3 Head

The model head is employed for the last stage of detection. An anchor box approach was used for the features, and a final output vector was produced that included class probabilities, objectness scores, and bounding boxes. In order to forecast the bounding box coordinates, class probabilities, and objectness score for each grid cell in the image, the head uses a three-layer output structure. The four coordinates, x, y, width, and height are predicted for each bounding box in the first output layer. Class probabilities are predicted for each bounding box in the second output layer, showing the likelihood that the identified item belongs to each class. For each predicted bounding box, the third output layer predicts an objectness score that indicates the possibility that an object is present within the box. The final object detection predictions are a product of the combination of these three levels of output. The architecture of our proposed model is shown in Figure 3.8. Our methodology focuses on capitalizing on the inherent strengths and improving the performance of YOLOv5 while maintaining its basic architecture. The main emphasis of our study revolves around the careful adjustment of hyperparameters and the customization of the model to more effectively address the distinct obstacles associated with identifying trash in natural settings. The commitment to enhancing the model's efficacy in the given environment is demonstrated through our proactive initiatives, which involve adjusting hyperparameters and selectively pretraining the model using a dedicated dataset focused on trash-related information.

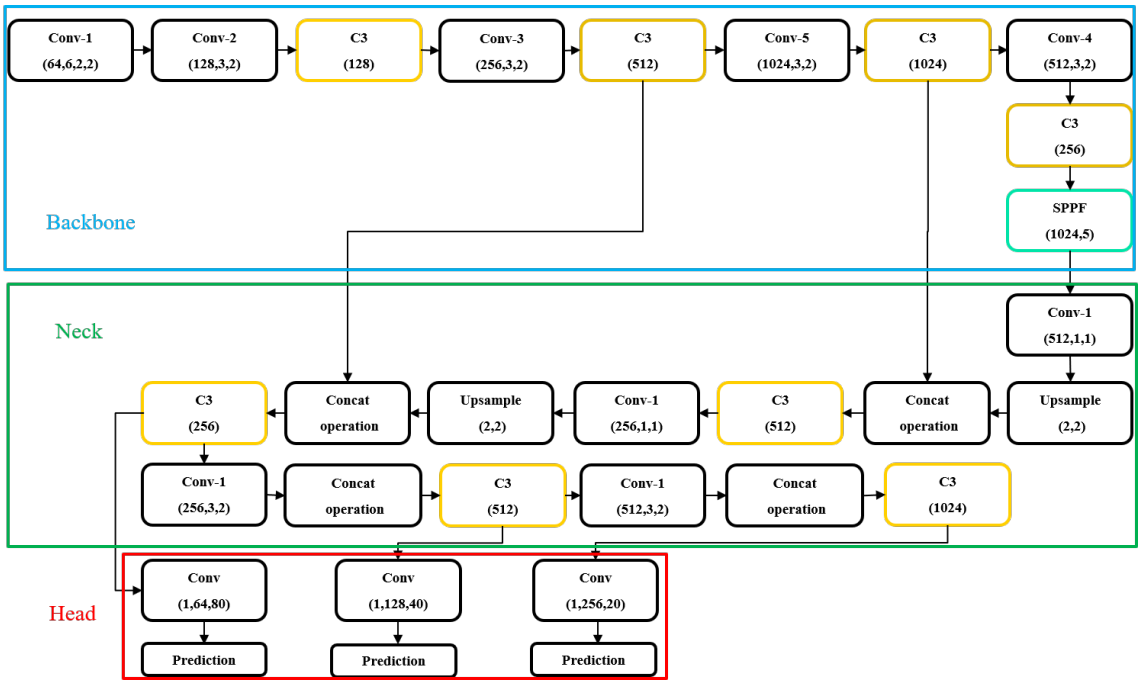

Figure 3.8. YOLOv5 model architecture

3.5.4 Transfer Learning

Transfer learning is a method used in machine learning to exploit the information obtained by pre-trained models on a task to improve the performance of a model being used for a task related to the original job. Transfer learning is very effective in object detection in terms of improving the performance of models by using the features learned from pre-trained models on a large-scale dataset such as ImageNet [39] or COCO [40]. Pre-training on large datasets has been a major driving force in the advancement of computer vision, with applications in fields as diverse as classification problems [41]–[44], detection problems [45]–[48] and many more. Due to the small size of our dataset, we relied on the YOLO models' weights as learned on the larger COCO dataset.

3.5.5 Activation Function

To better comprehend the intricate relationship between input and output variables, activation functions add non-linearity to the networks. Sigmoid, Tanh, ReLU, ELU, Swish, and Mish are just a few examples of well-liked activation functions [49]. Sigmoid-weighted Linear Unit [50] (SiLU) is utilized in YOLOv5's most layers. SiLU, like ReLU, is a non-monotonic activation function, but its curve is smoother, and it also addresses the critical problem of dying ReLU. The SiLU function, much like the sigmoid function, converts any value that is inputted into a number that is between 0 and 1. Equation 3.1

depicts our working definition of SiLU.

$$SiLU(I) = I \times sigmoid(I) \tag{3.1}$$

where I denotes the input variable, and sigmoid is defined as:

$$sigmoid(I) = \frac{1}{1 + e^{-I}} \tag{3.2}$$

The SiLU activation function has been employed in conjunction with convolutional operations within the hidden layers. The Sigmoid activation function has been utilized in conjunction with convolution operations within the output layer. In the domain of object detection or image classification tasks, it is customary for the output layer of the model to estimate the probability of object presence in various regions of the input image. The problem is commonly framed as a binary classification task, wherein each region is categorized as either "object present" or "object absent". The Sigmoid activation function is highly suitable for binary classification tasks due to its ability to compress output values within the range of 0 and 1. This enables the model to interpret the resulting output as a numerical value that represents the probability of an object's presence in each specific region of the image.

3.6 Training Process

The experiments were performed on an AMD Ryzen 5 3600 equipped with 16 GB of RAM. The software version is built on Pytorch 1.13.1 and Python 3.9.16. Because larger YOLO models like YOLOv5x, YOLOv5l, and YOLOv5m require more Memory and GPU, the Google colab pro version has been used for those models. In accordance with the accessible resources, Google Colab Pro allows 32 GB of RAM and 13 GB of Tesla T4 storage.

We have conducted several experiments on our dataset. To get with, we were just concerned with the dataset pertaining to Bangladesh. Data collected from the environment in Bangladesh is fairly difficult to interpret because of the abundance of different categories. Furthermore, the vast majority of data from the Bangladeshi dataset were collected from a distance of five feet or farther. Because of the constraints imposed by the available hardware, the resolution of the data is not nearly as high as it should be. Training will need to be done for a longer period of time if the resolution is increased. When used in object recognition, high-resolution images produce superior results due to the comprehensive

feature map extraction made possible by the images. There are 1283 images and 6178 annotations in our Bangladeshi dataset. We divided our dataset into 80:10:10 pieces, which means that out of 1283 images, 1026 were chosen for training, 128 were chosen for validation, and 129 were chosen for the test set. Following that, we expanded our dataset by merging the data we acquired from openlittermap. We have 4418 images and 9837 annotations for them all together. After the right split, we had 3524 images for training, 442 for testing, and 442 for validation.

We evaluated a variety of parameters to see how well our preferred model performed. We have employed the stochastic gradient descent (SGD) method for every experiment as an optimizer. We reduced the image size of our dataset and other datasets used in our experiments to 640×640. Since larger images have the potential to provide more feature information for object detection, we have opted for an input image size of 640×640. Choosing a larger size would improve accuracy, but it would come at a higher computational cost. In all of our experiments, a total of 100 epochs were utilized. The reason behind employing a uniform epoch number across all models was determined through a meticulous evaluation of various factors, encompassing computational resources, model convergence, and generalization performance. Considering the limitations imposed by our computational resources and time limitations, we made the decision to use a consistent and uniform number of epochs. This choice was made in order to ensure that fair and consistent comparisons could be made between the different models. The selected hyperparameters for our experiments are listed in Table 3.6.

Table 3.6. Hyperparameter values for our experiments

Model	Image size	Learning rate	Epochs	Batch size
YOLOv5s	640×640	0.00334	100	12
YOLOv5s	640×640	0.01	100	12
YOLOv5s	640×640	0.00334	100	24
YOLOv5s	640×640	0.00334	100	32
YOLOv5x	640×640	0.00334	100	12
YOLOv5x	640×640	0.01	100	12
YOLOv5l	640×640	0.00334	100	12
YOLOv5m	640×640	0.00334	100	12

3.7 Conclusion

This chapter explores the extensive procedure of categorizing trash through the utilization of deep neural networks. The justification for selecting a variety of deep learning algorithms for this task is given significant emphasis. The following chapter will analyze the results obtained from the experiments done on the proposed deep neural network structures.

Chapter 4

Experimental Result Analysis

4.1 Introduction

This chapter initiates an investigation into the performance evaluation of the YOLOv5 models that have been specifically designed for the purpose of trash classification. The aim of this study is to assess the effectiveness of these models in accurately classifying different categories of trash. In order to obtain a thorough comprehension of the model's performance, we do an in-depth examination of the system design, spanning various accuracy measures like precision, recall, and f1-score. Furthermore, we explore factors pertaining to the duration of inference and the computing efficiency, which is quantified in terms of GFLOPS. Our experiments were split up into these two distinct parts. In the first part of this section, we look at all the classes and conduct experiments to identify different kinds of trash. Experiments will be carried out in the second segment, during which we will combine all classes into a single trash category.

4.2 System Configuration

The system configuration for all of our experiments is outlined in Table 4.1.

Table 4.1. System configuration

Name	**Details**
Processor	AMD Ryzen 5 3600 Processor
RAM	16 GB
Operating System	Ubuntu 20.04
Library and Framework	Pytorch 2.0
Programming Language	Python 3.7
IDE	Spyder
Others	Google Colab Pro (32 GB)

4.3 Loss Function and Evaluation Metrics

In order to train the model for object detection, the YOLOv5 model employs a mix of different loss functions [51]. The total loss function may be stated in the following format:

$$Loss_{total} = Loss_{box} + Loss_{class} + Loss_{object} \tag{4.1}$$

where the bounding box regression loss function, the classification loss function, and the confidence loss function, respectively, are referred to as $Loss_{box}$, $Loss_{class}$, and $Loss_{object}$.

To evaluate how well the model is doing, the YOLOv5 object detection algorithm uses a few different assessment criteria [52]. These metrics consist of the following:

- Intersection over Union: The evaluation metric of Intersection over Union (IoU) is widely employed in object detection tasks to quantify the degree of similarity between two bounding boxes. It is calculated by dividing the area of the union of the predicted and ground-truth bounding boxes by the area of the intersection. The IoU measure runs from 0 to 1, with 1 indicating a complete overlap between anticipated and ground-truth bounding boxes. The following formula can be used to get the IoU:

$$IoU(P, Q) = \frac{P \cup Q}{P \cap Q} \tag{4.2}$$

  where P is the set of the predicted areas, and Q is the set of the actual areas.
- Precision: Precision is defined as the fraction of predictions that turn out to be correct relative to the total number of positive predictions. The formula for precision is:

$$precision = \frac{TP}{TP + FP} \tag{4.3}$$

  where TP and FP indicate the true positive and false positive, respectively.
- Recall: Recall, also known as sensitivity or true positive rate (TPR), is the percentage of positive instances that the model properly recognized. The formula for the recall is:

$$recall = \frac{TP}{TP + FN} \tag{4.4}$$

  where FN is the number of false negatives, i.e., positive instances projected as negative.
- F1-score: The F1-score is a combined measurement of both precision and recall, and it is calculated as the harmonic mean of these two measures. The F1-score strikes a compromise between accuracy and recall, making it beneficial when it is necessary to have both high precision and strong recall. The following is the formula for the

F1-score:

$$f1\text{-}score = \frac{2 \times (precision \times recall)}{precision + recall} \tag{4.5}$$

- Mean Average Precision (mAP): The Mean Average Precision measure is a standard method of comparing different object detection algorithms. It evaluates the model's accuracy on an average basis, taking into account various confidence levels and thresholds. YOLOv5 calculates the mAP using the IoU of anticipated and ground-truth bounding boxes. The mAP is obtained by doing independent calculations for each object class and then taking the average of those results for all classes, as shown in Equation 4.6.

$$mAP = \frac{1}{N} \sum_{w=1}^{N} \frac{TP(w)}{TP(w) + FP(w)} \tag{4.6}$$

  where N is the number of classes, TP(w) is true positive instances, and FP(w) is false positive instances. Hence, the value of the metric $mAP_{0.5:0.95}$ reflects the mAP over a range of IoU thresholds that goes from 0.5 to 0.95 with increments of 0.05. Similarly, $mAP_{0.5}$ defines mAP for an IoU that is more than 0.5, and $mAP_{0.75}$ represents mAP for an IoU that is greater than 0.75. In our work, we decided to use $mAP_{0.5}$ as the primary criterion for assessment, in addition to precision, recall, and F1-score.
- Giga Floating-Point Operations per Second (GFLOPs): The computing performance of a neural network model is measured in terms of GFLOPs. It is the maximum rate at which the model's GPU or CPU can execute floating-point calculations. When comparing neural network models, the GFLOPs metric can be helpful for quantifying the difference in computational efficiency. Generally, a model with a higher GFLOPs value is more computationally efficient and can process input data faster than one with a lower GFlops value. Even though a higher GFLOPs value might mean that a model is performing better in terms of computation time, however, it does not mean that it can work faster or provide more accurate results. We have considered a number of performance metrics, such as accuracy and inference time, in addition to GFLOPs.
- Inference Time: The inference time is measured as the amount of time it takes for the model to analyze an input image and provide a prediction. There are a lot of variables that can affect this time, including the number of objects to be identified, the size and complexity of the input image, and the hardware utilized for inference.

4.3.1 Results on our dataset

First, we concentrated our focus on the data collected from Bangladesh. We selected different batch sizes and learning rates for our experiments. The outcomes of our experiments are displayed in Table 4.2. There are a total of four different YOLOv5 models

shown in this table. Using a batch size of 12 and a learning rate of 0.01, the YOLOv5x model achieved an mAP of 33.3% for IoU 50. This result is greater by 31.7% compared to YOLOv5m, 16.8% compared to YOLOv5l, and 14.8% compared to YOLOv5s. On the other hand, as compared to YOLOv5s, the inference time of the YOLOv5x model is painfully sluggish. Following the implementation of TTA with the YOLOv5x model, we recognized an improvement in mAP, precision, and f1-score, as well as a reduction in recall and inference time. We examined the performance of two distinct learning rates to see which one is superior, and we observed that the YOLOv5 model performs admirably with a learning rate of 0.01 when applied to our Bangladeshi dataset.

Experiments with various different variants of YOLO [53], [54], and Faster R-CNN have also been carried out, as can be shown in Table 4.3. According to the table, we can conclude that the YOLOv5 model performs better than the YOLOv6, YOLOv8, and Faster R-CNN models on our dataset. Table 4.4 lists parameter counts for our experimental models. As can be observed from the table, the Faster R-CNN algorithm requires the maximum number of parameters, while the YOLOv5s algorithm requires the lowest number of parameters. The comparison of training times between different models is depicted in Figure 4.1, which highlights the variations in training duration. The figure indicates that smaller models such as YOLOv5s (22.92min), YOLOv5m (43.32min), YOLOv6s (47.16min), and YOLOv8s (26.34min) exhibit less training times, whereas larger models like YOLOv5l (69.54min) and YOLOv5x (121.44min) demonstrate a gradual increase in training time. The Faster R-CNN model necessitated a greater amount of time (370min) for training in comparison to alternative models. In contrast, Figure 4.2 presents a visual representation of the inference times for the different models, offering valuable insights into their respective execution efficiencies during the inference process. The figure illustrates that smaller models exhibit lower inference times (YOLOv5s - 6.9ms, YOLOv6s - 10.49ms, YOLOv8s - 22.3ms), whereas larger models such as YOLOv5l (28.9ms) and YOLOv5x (59.1ms) demonstrate higher inference times than other models. The Faster R-CNN model exhibits a significantly longer inference time (102ms) than the YOLOv5x model. Models with faster inference times indicate greater efficiency, rendering them more appropriate for real-time applications or scenarios. As a result, we will continue the rest of the experiments with the YOLOv5 model. The representation of mAP on our dataset for the YOLOv5x model is shown in Figure 4.3. A comparison of SGD with several other optimizers has also been carried out, and as shown in Figure 4.4, SGD outperforms in this regard. The properties of the dataset and the peculiarities of the training procedure may influence the choice of the Stochastic Gradient Descent (SGD) optimizer over ADAM and ADAMW. SGD is a simple optimization technique that uses the gradient of the loss function with respect to the parameters to update model parameters. SGD can be more effective in discovering optimal solutions in circumstances where the dataset is not excessively huge, and the optimization

Table 4.2. Experimental results on the Bangladeshi dataset

Model	Lr	mAP@50	mAP@50:95	Precision	Recall	F1-Score	GFLOPs	Inference (ms)	Bs
YOLOv5x	0.00334	31.1	13.6	52.8	33.4	40.9	203.9	53.7	12
YOLOv5x + TTA	0.00334	31.1	13.8	61.7	32.8	42.8	203.9	57.3	12
YOLOv5x	0.01	33.3	14.6	49.9	38.8	43.6	203.9	59.1	12
YOLOv5x+TTA	0.01	**34.3**	**15.1**	**51.6**	**38**	**43.7**	**203.9**	**65.7**	12
YOLOv5m	0.00334	25.3	11	57.4	27.1	36.8	48.0	15.7	12
YOLOv5m+TTA	0.00334	25.3	11.5	56.3	31.6	40.4	48.0	18.0	12
YOLOv5l	0.00334	28.5	13.2	62.1	30.8	41.1	107.8	28.9	12
YOLOv5l+TTA	0.00334	30.6	14.2	61.8	32.9	42.9	107.8	27.9	12
YOLOv5s	0.00334	14.8	6.54	66.9	18.1	28.4	15.8	6.9	12
YOLOv5s+TTA	0.00334	14.9	6.51	67.3	18.7	29.2	15.8	13.9	12
YOLOv5s	0.00334	14.7	6.44	67.1	18.2	28.6	15.8	6.7	24
YOLOv5s	0.00334	15.3	6.72	66.9	18.3	28.7	15.8	6.4	32
YOLOv5s	0.01	29	12.4	51.3	29.2	37.2	15.8	6.8	32
YOLOv5s	0.01	27	11.9	38.4	29.9	33.6	15.8	6.6	12

Lr = Learning rate, **Bs** = Batch size, **TTA** = Test Time Augmentation.

Table 4.3. Comparison with other models on our dataset

Model	mAP@50
YOLOv5x	**34.3**
YOLOv6s	32.1
YOLOv8s	31.9
Faster R-CNN	27.3

landscape is not extremely complex. In addition to SiLU, we compared several other activation functions, such as ReLU, LeakyReLU, and GELU [55]. Out of these, SiLU performed relatively better, as can be shown in Figure 4.5.

In Figure 4.6, we have displayed both the original image as well as the images that were detected for our test data. We can see that the YOLOv5x model accurately identified the trash types and properly labeled them with a confidence score of 75% and 89%, respectively, in Figure 4.6(a). The confidence score for YOLOv5's model is lower than that of YOLOv5x, despite the fact that it correctly identified trash categories. While YOLOv5m performed similarly to YOLOv5s, YOLOv5l could not correctly identify some of the categories in the image. In Figure 4.6(b), the YOLOv5x model detected ten categories but incorrectly categorized paper as plastic. YOLOv5s correctly recognized the paper; however, tiny pieces of trash, such as cigarette butt and medicine, were not recognized correctly. The YOLOv5l and YOLOv5m models also failed to identify some trash; hence, the YOLOv5x model performed better than the other models in both cases.

Table 4.4. Number of parameters of our experimental models

Model	Number of parameters
YOLOv5s	7.04M
YOLOv5m	20.9M
YOLOv5l	46.2M
YOLOv5x	86.2M
YOLOv6s	17.2M
YOLOv8s	11.1M
Faster R-CNN	165.2M

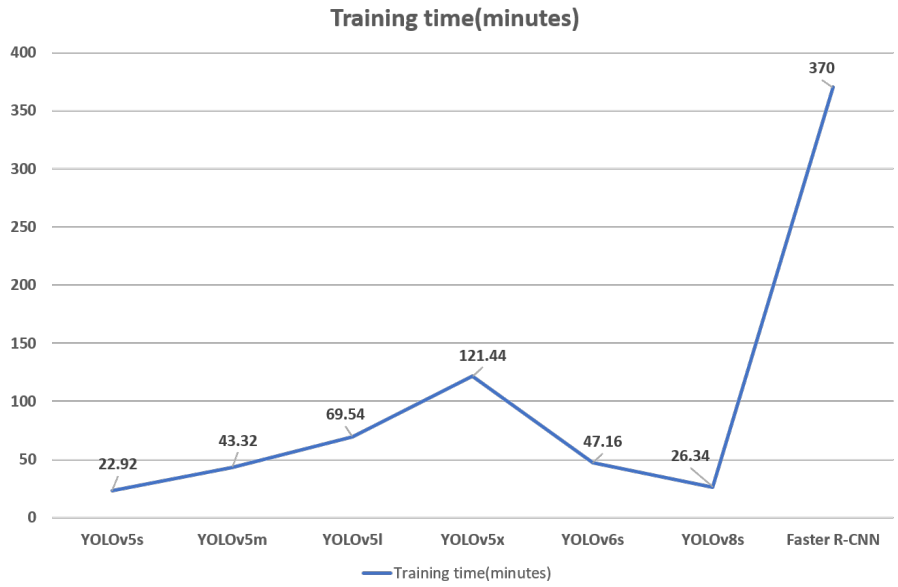

Figure 4.1. Training time comparisons for different models

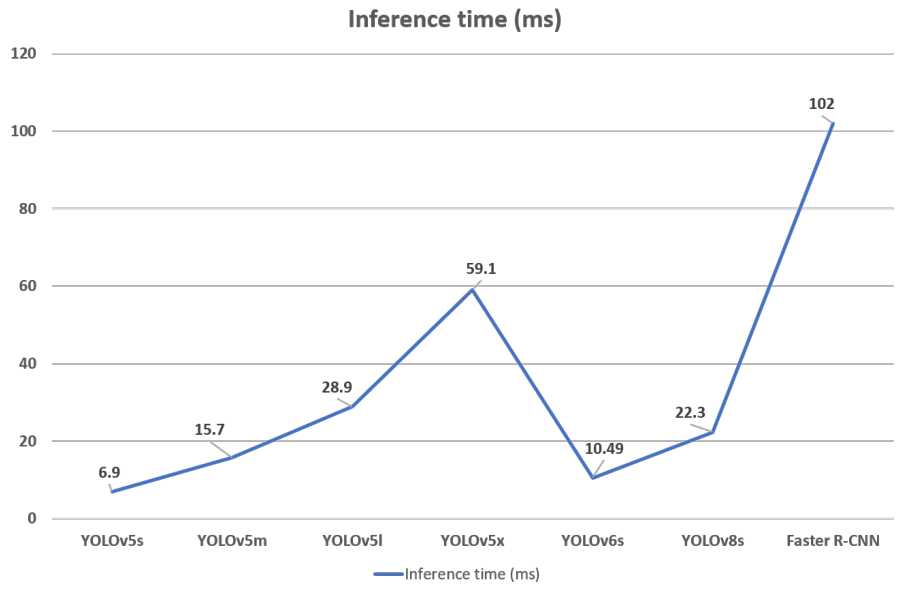

Figure 4.2. Inference time comparisons for different models

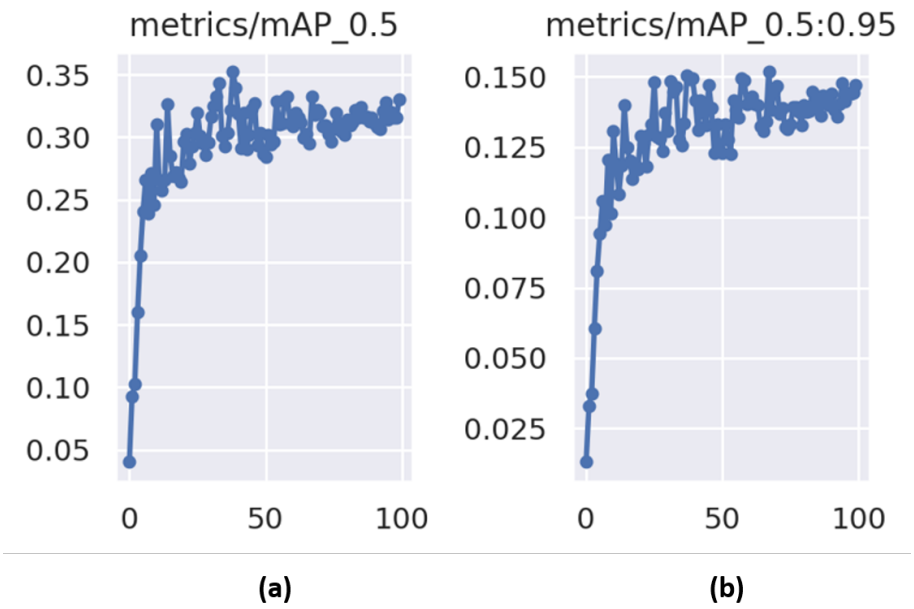

Figure 4.3. mAP for YOLOv5x on Bangladeshi dataset at (a) IoU 0.50, (b) IoU 0.50:0.95

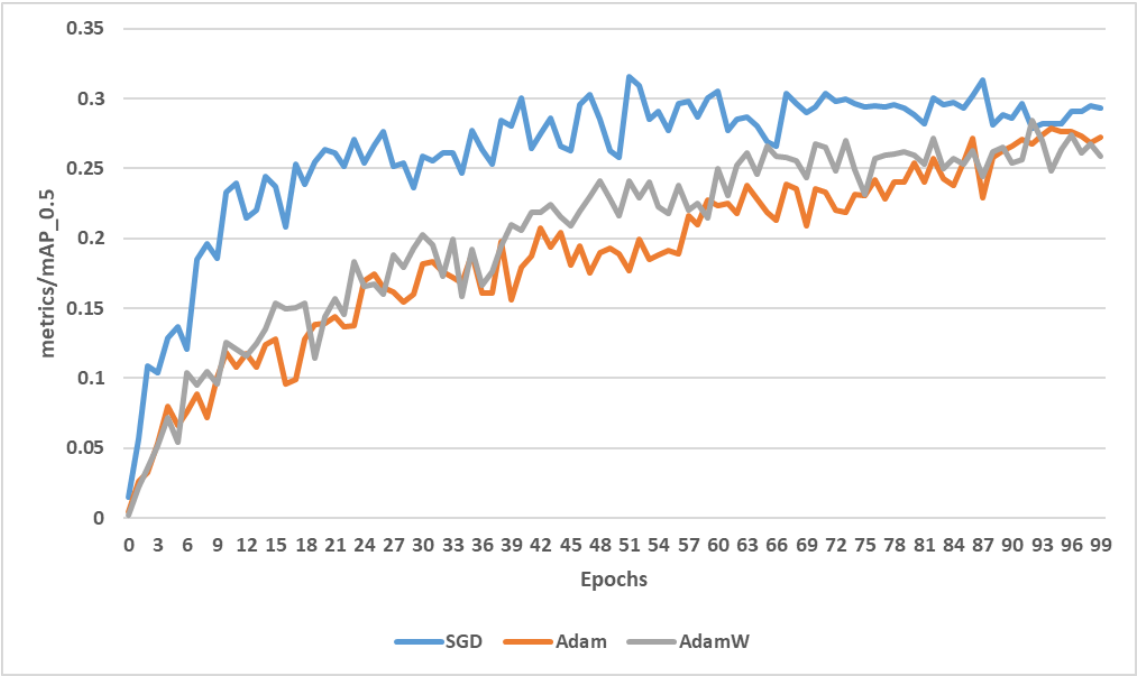

Figure 4.4. Comparison of different optimizers on our dataset

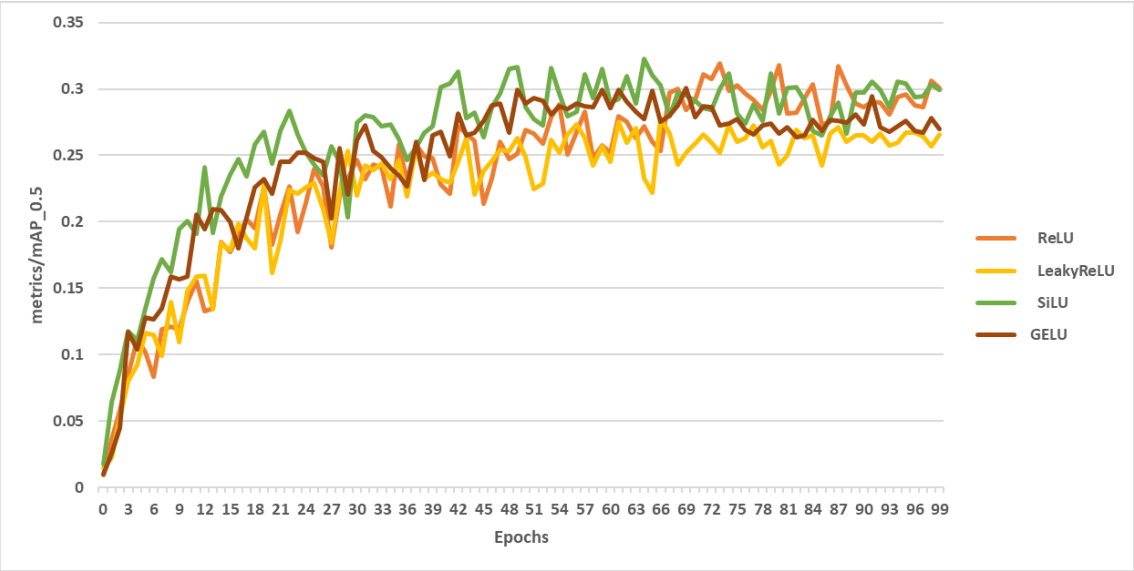

Figure 4.5. Comparison of different activation functions on our dataset

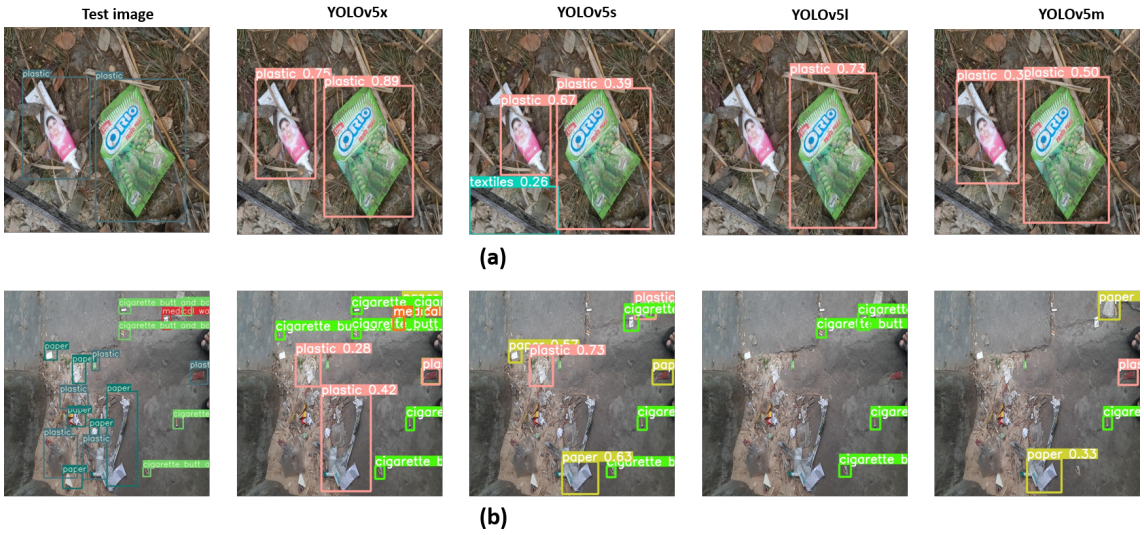

Figure 4.6. Detection results on sample test data (a) 2 instances, (b) 17 instances

4.3.2 Results on extended dataset

We expanded our dataset derived from openlittermap to ensure the diversity of the various categories. We put the four YOLOv5 models through their paces by adjusting the hyper-parameters, and the outcomes are presented in Table 4.5. We can see from the table that YOLOv5l accomplished an mAP of 45.2%, having a learning rate of 0.00334 and a batch size of 12. The accuracy is superior to that of the YOLOv5x model, which achieved the best result in the experiment that we had conducted previously. The F1-score and inference time are significantly improved compared to the other models utilized in this experiment.

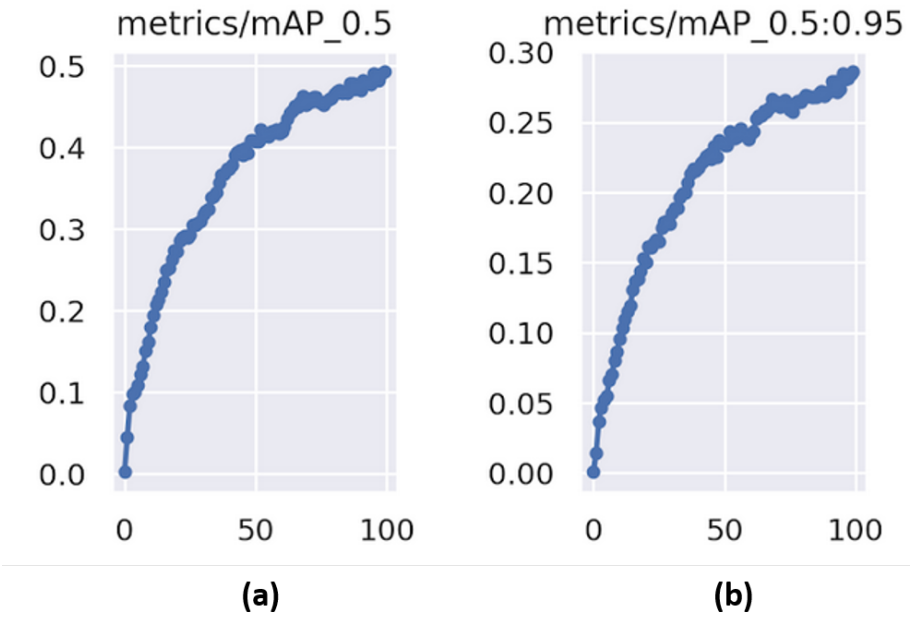

Figure 4.7. mAP for YOLOv5l on extended dataset at (a) IoU 0.50, (b) IoU 0.50:0.95

Minor improvements were seen in mAP@0.50, F1-score, and inference time after TTA was applied. The mAP of the YOLOv5l model is shown in Figure 4.7.

Figure 4.8 shows the experimental results for two sample test images taken from our extended dataset, executed on YOLOv5 models. We have shown two cases, one with four instances, as in Figure 4.8(a), and the other with ten instances, as in Figure 4.8(b). In Figure 4.8(a), every model, with the exception of YOLOv5x, identified three instances. When compared to other models, the confidence score assigned to each piece of trash in YOLOv5's model is significantly greater. In addition, YOLOv5s and YOLOv5m are the only models that accurately identified two separate instances of paper. In Figure 4.8(b), the YOLOv5x model identifies eight out of ten instances of the problem. The YOLOv5l algorithm found seven instances, and it had a high confidence level in detecting tissue paper.

4.3.3 Results on existing datasets

We have used two existing datasets to demonstrate the efficacy of our chosen models. TACO has sixty distinct classes, while Plastopol has only one. In Table 4.6, we can see the results of all the tests conducted on TACO. At a learning rate of 0.01 and a batch size of 12, the YOLOv5l model produced an mAP of 25.5% at IoU@50. While the YOLOv5x model's output was comparable to that of the YOLOv5l model, the latter model performed better in terms of both the f1-score and the inference time. The mAP of the YOLOv5l model

Table 4.5. Experimental results on the extended dataset (Bangladeshi dataset + openlitter-map)

Model	Lr	mAP@50	mAP@50:95	Precision	Recall	F1-Score	GFLOPs	Inference (ms)	Bs
YOLOv5x	0.00334	43.9	25.9	56.6	42.9	48.8	203.9	58.9	12
YOLOv5x + TTA	0.00334	44.2	26.2	64.2	44.4	52.4	203.9	54.0	12
YOLOv5x	0.01	42.2	24.9	56	42	48	203.9	56.7	12
YOLOv5x+TTA	0.01	42.7	25.5	52.3	46.1	49	203.9	50.7	12
YOLOv5m	0.00334	40.2	24.4	62.5	38.6	47.7	48	15.4	12
YOLOv5m+TTA	0.00334	40.1	24.8	58.2	41.2	48.2	48	17.1	12
YOLOv5l	0.00334	45.2	27.6	67.8	44.9	54	107.8	29.0	12
YOLOv5l+TTA	0.00334	**45.4**	**27.5**	**70.1**	**45.7**	**55.3**	**107.8**	**27.8**	12
YOLOv5s	0.00334	30	18.5	67.2	29.7	41.1	15.8	6.4	12
YOLOv5s+TTA	0.00334	31.8	20	67.5	30.7	42.2	15.8	8.8	12
YOLOv5s	0.00334	30.4	18.5	69.9	31.4	43.3	15.8	6.3	24
YOLOv5s	0.00334	30.7	18.8	69.9	29.9	41.8	15.8	6.2	32
YOLOv5s	0.01	41.2	23.1	37.8	51	43.4	15.8	6.3	32
YOLOv5s	0.01	38.8	22.3	42.8	39.1	40.8	15.8	6.2	12

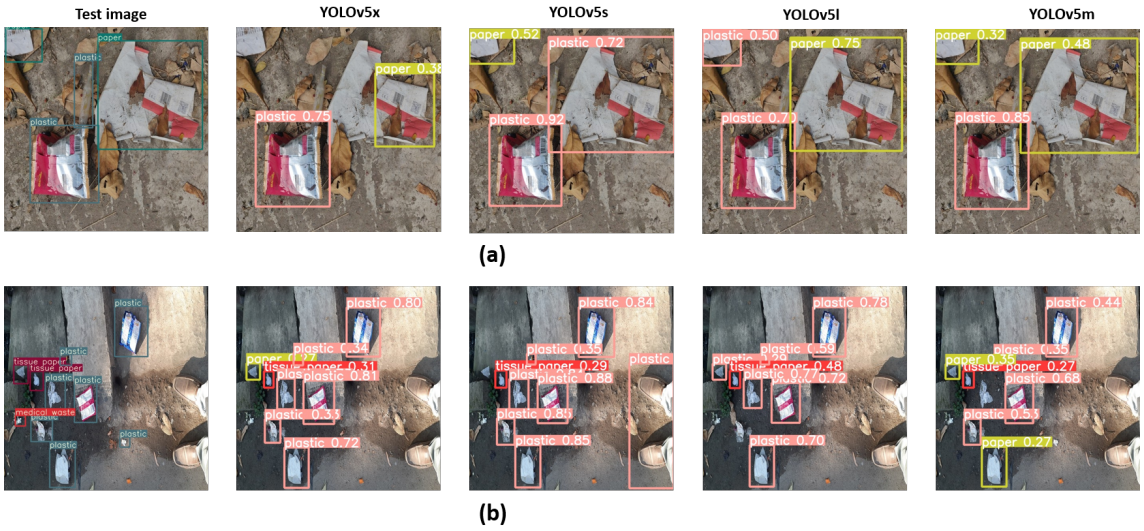

Figure 4.8. Detection results on test data of our extended dataset (a) 4 instances, (b) 10 instances

improved with TTA but at the cost of more inference time. We also tried the YOLOv5x model with a learning rate of 0.00334, but the results were unsatisfactory.

We have also compared our findings to those of previously published works using TACO datasets shown in Table 4.7. While previous authors have reported an mAP of 16.2% and 17.6% for their work with 7 and 10 classes, respectively, our experiments with 60 classes yielded an mAP of 25.5%.

4.3.3.1 Single class detection

Some authors [13], [25], [56] have worked on single-class detection problems in which the goal is to determine whether trash is present in an image. For single-class detection, we employed TACO and PlastOpol. For the TACO dataset, the results are shown in Table 4.9. According to the data in the table, the YOLOv5x model with a batch size of 12 and a

Table 4.6. Experimental results on the TACO dataset (60 classes)

Model	Lr	mAP@50	mAP@50:95	Precision	Recall	F1-Score	GFLOPs	Inference (ms)	Bs
YOLOv5x	0.00334	9.7	7.68	69	9.22	16.2	205	43.0	12
YOLOv5x	0.01	22.5	18.9	27.6	31.3	29.3	205	42.2	12
YOLOv5x+TTA	0.01	25.2	20.5	45.2	24.5	31.7	205	48.9	12
YOLOv5m	0.01	20.5	16.1	28.1	24.5	26.1	48.6	15.5	12
YOLOv5m+TTA	0.01	22	17.3	37.6	27	31.4	48.6	18.0	12
YOLOv5l	0.01	22.9	18.1	37.2	27.9	31.8	108.7	28.9	12
YOLOv5l+TTA	0.01	**25.5**	**19.9**	**46.3**	**25.4**	**32.8**	**108.7**	**42.2**	12
YOLOv5s	0.01	15.8	10.7	36.2	20.5	26.1	16.3	8.7	12
YOLOv5s+TTA	0.01	17.6	12.2	50.1	18.8	27.3	16.3	27.5	12
YOLOv5s	0.01	17.7	12.1	48.6	18.7	27	16.3	6.6	24
YOLOv5s	0.01	18.4	13.3	43.6	21.2	28.5	15.8	7.8	32

Table 4.7. Comparison of TACO datasets (multiple classes)

Author	Dataset	mAP@50
Majchrowska et al. [56]	Extended TACO(7 classes)	16.2
Pedro et al. [13]	TACO (10 classes)	17.6
Ours	TACO (60 classes)	**25.5**

Table 4.8. Fold selection of PlastOpol dataset

Model	Fold	mAP@50	F1-score
YOLOv5x	fold 1	84.7	80.3
	fold 2	85.4	77.9
	fold 3	**87.8**	**83.9**
	fold 4	85.7	77.6
	fold 5	85.3	80.3

Table 4.9. Experimental results on the TACO dataset (one class)

Model	Lr	mAP@50	mAP@50:95	Precision	Recall	F1-Score	GFLOPs	Inference (ms)	Bs
YOLOv5x	0.00334	**61.4**	**47.6**	**84.1**	**51.3**	**63.7**	**203.8**	**40.5**	12
YOLOv5x+TTA	0.00334	60.6	47.3	78.7	51.1	61.9	203.8	50.0	12
YOLOv5m	0.00334	57.4	42.7	82.8	49.1	61.6	47.9	13.8	12
YOLOv5m+TTA	0.00334	56.7	42.7	78.5	50.4	61.3	47.9	34.2	12
YOLOv5l	0.00334	60.3	46.1	72.9	54.1	62.1	107.6	22.4	12
YOLOv5l+TTA	0.00334	59.6	46	67	56.2	61.1	107.6	37.0	12
YOLOv5s	0.00334	55	38.5	68.7	50.1	57.9	15.8	7.1	12
YOLOv5s+TTA	0.00334	55.4	39.1	77.8	45.9	57.7	15.8	26.3	12
YOLOv5s	0.00334	54.7	37.9	68.6	49.9	57.7	15.8	6.0	24
YOLOv5s	0.00334	54.4	38	68.4	49.1	57.1	15.8	6.4	32
YOLOv5s	0.01	52.6	37.7	72.6	48.4	58	15.8	7.3	12
YOLOv5x	0.01	58.9	46.2	77.8	52.3	62.5	203.8	39.5	12

Table 4.10. Experimental results on the PlastOpol dataset - fold 3 (one class)

Model	Lr	mAP@50	mAP@50:95	Precision	Recall	F1-Score	GFLOPs	Inference (ms)	Bs
YOLOv5x	0.00334	87.8	74.7	88.2	80	83.9	53.4	53.4	12
YOLOv5x+TTA	0.00334	**88.4**	**74.8**	**86.9**	**81.2**	**83.9**	**203.8**	**49.7**	12
YOLOv5m	0.00334	84.6	69.3	84.7	77.6	80.9	47.9	15.4	12
YOLOv5m+TTA	0.00334	85	69.7	86.8	74.9	80.4	47.9	17.3	12
YOLOv5l	0.00334	86.8	72.8	88	77.8	82.5	107.6	27.7	12
YOLOv5l+TTA	0.00334	86.9	73	86.5	79	82.5	107.6	26.5	12
YOLOv5s	0.00334	78.2	60.2	82	69.6	75.2	15.8	6.1	12
YOLOv5s+TTA	0.00334	79.8	61.7	82.2	71.1	76.2	15.8	7.9	12
YOLOv5s	0.01	83.4	67.7	90.7	73.7	81.3	15.8	6.0	32
YOLOv5s	0.01	83.3	68.1	86.4	76.8	81.3	15.8	6.1	12

Table 4.11. Single class experiment comparison with the TACO dataset

Author	Dataset	Model	mAP@50
Majchrowska et al. [56]	Extended TACO	EfficientDet D2	55.7
Pedro et al. [13]	TACO	Mask R-CNN	26.2
Cordova et al. [25]	TACO	YOLOv5s	54.7
Ours	TACO	YOLOv5s	55
Ours	TACO	YOLOv5m	**57.4**

Table 4.12. Single class experiment comparison with the PlastOpol dataset

Author	Dataset	Model	mAP@50
Cordova et al.[25]	PlastOpol	YOLOv5x	84.9
Ours	PlastOpol	YOLOv5x	**88.4**

Table 4.13. Experimental results on our extended dataset (one class)

Model	Lr	mAP@50	mAP@50:95	Precision	Recall	F1-Score	GFLOPs	Inference (ms)	Bs
YOLOv5x	0.00334	83.2	50.1	81.7	74.4	77.8	203.8	53.7	12
YOLOv5x+TTA	0.00334	83.7	50.6	80.5	74.1	77.1	203.8	49.7	12
YOLOv5m	0.00334	82.2	49.3	79.6	76	77.7	47.9	16.2	12
YOLOv5m+TTA	0.00334	83.2	50.2	81	74.5	77.6	47.9	16.5	12
YOLOv5l	0.00334	83.4	49.9	81	75.2	77.9	107.6	28.0	12
YOLOv5l+TTA	0.00334	**84.4**	**50.9**	**81.6**	**75.2**	**78.2**	**107.6**	**27.4**	12
YOLOv5s	0.00334	80.1	46.5	81	69.4	74.7	15.8	6.2	12
YOLOv5s+TTA	0.00334	81.2	48.4	80.8	70.9	75.5	15.8	9.5	12
YOLOv5s	0.01	78.7	43.9	79.2	71.8	75.3	15.8	6.0	12
YOLOv5s	0.01	78.9	43.7	78.7	72.5	75.4	15.8	5.9	32

learning rate of 0.00334 produced an mAP of 61.4%. Since the PlastOpol dataset was distributed in five subsets, we applied the YOLOv5x model to each subset to see which gave the best results, as shown in Table 4.8. According to the table, fold three produces the highest mAP and f1-score, which we used to determine the optimal fold. Our PlastOpol experiment results are summarized in Table 4.10. We used a batch size of 12 and a learning rate of 0.00334, and we obtained an mAP of 87.8%. The mAP increases to 88.4% after the use of TTA. In addition, we tried out the YOLOv5s model with various batch sizes and found that it produced results that were superior to those produced by the same model with a batch size of 12. In addition, Table 4.11 and 4.12 provide a comparison with existing datasets. Our selected YOLOv5 models performed better than other models for the TACO and PlastOpol datasets, as shown in the tables. For the TACO dataset, Cordova et al. [25] obtained an mAP of 54.7% using the YOLOv5s model, whereas we were able to obtain an mAP of 55.0% using the same model. In the evaluation of the TACO dataset, a comparison was made between Mask R-CNN, EfficientDet, and YOLOv5. The results indicated that YOLOv5 exhibited a higher accuracy rate of 57.4%, surpassing the accuracy rates of Mask R-CNN (26.2%) and EfficientDet (55.7%). This information can be found in Table 17. The observed variation in performance can be ascribed to multiple factors, encompassing the dissimilarity in architectural design between the models, the specific attributes of the TACO dataset, the efficacy of YOLOv5's single-stage methodology,

and its capacity to effectively handle objects of varying scales and aspect ratios through anchor-based detection. Furthermore, YOLOv5 could have potentially derived advantages from the implementation of efficient data augmentation techniques, meticulous tuning of hyperparameters, and optimal allocation of computational resources throughout the training process. For the PlastOpol dataset, Cordova et al. [25] attained an mAP of 84.9% using the YOLOv5x model, whereas we were able to acquire an mAP of 87.8% using the same model as shown in Table 4.12. The activation function utilized in the experiments conducted by Cordova et al. [25] was not explicitly specified. Nevertheless, our investigation of the PlastOpol dataset revealed that employing the SiLU activation function within the convolutional layer yielded superior accuracy in contrast to their findings. Furthermore, a distinct learning rate was employed, resulting in enhanced performance. The significance of architectural choices, activation functions, and hyperparameters is underscored by these findings, as they have the potential to exert a substantial influence on the performance of a model when applied to particular datasets. In addition, we performed experiments on our extended dataset. As shown in Table 4.13, we achieved an mAP of 83.4% using the YOLOv5l model and a batch size of 12. With the implementation of TTA, the mAP rises to 84.4%, and the f1-score climbs to 78.2%.

4.3.4 Discussion

We applied our selected model to our dataset in addition to two preexisting datasets in an effort to identify trash in the wild. Six different types of detection tasks are summarized in Table 2.1. Three of the datasets were not made available to the public. Experiments were carried out using publicly accessible datasets that were sufficiently complicated and suited the scope of our research. However, we were unable to incorporate some datasets, such as those pertaining to marine debris or a straightforward indoor setting, into our studies. In addition, there are datasets that have refrained from making their annotation files available to the broader public. As a consequence of this, there is a disparity between our mAP and the outcomes of the other studies presented in Table 2.1. Although we made advancements in enhancing the reported outcomes of certain models, as evidenced by our study's comparison to Cordova et al.'s previous research on litter [25], we encountered difficulties in attaining similar results with alternative datasets. The potential cause for the observed variability in performance may be ascribed to the utilization of Transfer Learning. Although Transfer Learning is a potent technique for harnessing the knowledge of pre-trained models on extensive datasets, it may not consistently produce optimal outcomes for particular datasets. The dataset we have employed pertains to the identification of trash in natural environments, and it possesses distinctive attributes and obstacles that may not have been comprehensively addressed in the pre-training phase. Consequently, it is plausible that the model's capacity to effectively adjust to our specific dataset might

have been constrained, thereby resulting in disparities in performance when compared to other datasets. In addition, the datasets utilized for evaluation demonstrated a wide range of environmental contexts. The dataset employed by Melinte et al. [29] was the TrashNet dataset, which comprised indoor images predominantly featuring individual instances of garbage. On the other hand, the dataset utilized in our study consists of data gathered from outdoor settings, thereby introducing a range of variables such as variations in lighting conditions, backgrounds, and object dimensions. As depicted in Figure 4.9(a) and Figure 4.9(b), we have taken into account two distinct datasets to facilitate a comparison between indoor and outdoor settings. For testing purposes, the indoor samples are sourced from the TrashNet dataset, while the outdoor samples are derived from the Bangladeshi dataset. The presence of diverse backgrounds can have a significant impact on object detection, as objects may be encountered in different levels of complexity, lighting conditions, and contextual components. The model's difficulty in adapting to complex circumstances, as exemplified by the mixing of sunlight and shadow in Figure 4.9(b), can be attributed to the obstacles posed by natural backgrounds. Likewise, the use of grassy backgrounds introduces an additional level of intricacy, hence increasing the level of difficulty in achieving precise object detection. The aforementioned characteristics have the potential to impact the performance of the model on our dataset. The intricacy of the dataset may be to blame for the mAP gap, whereas the data from natural environments are the primary focus of our investigation.

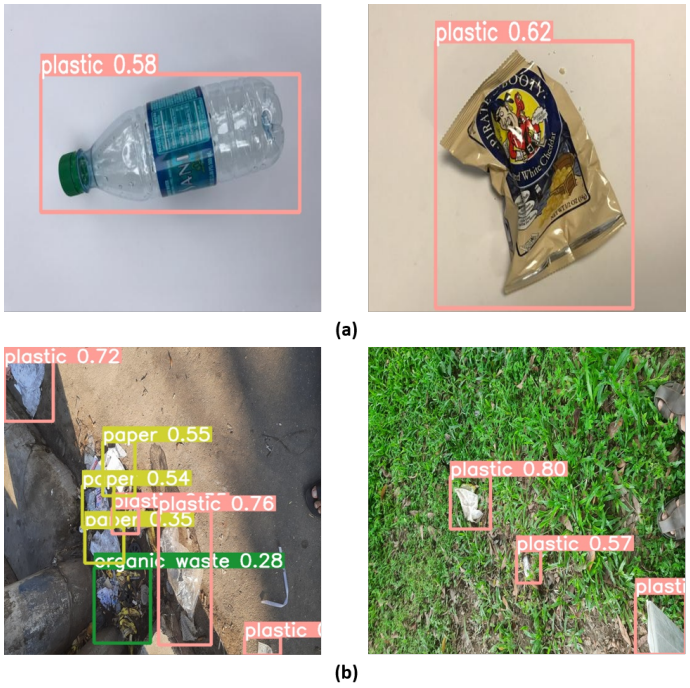

Figure 4.9. Comparison of Indoor and Outdoor Environments across two different datasets. The images in (a) depict indoor environments sourced from the TrashNet dataset, whilst the images in (b) exhibit outdoor situations derived from the Bangladeshi dataset.

4.4 Conclusion

The efficacy of different deep neural network architectures in our trash classification system was thoroughly evaluated through an extensive set of experiments. The YOLOv5 model consistently shows a high level of accuracy in the classification of a wide range of trash objects, which is in line with the goals we set out to achieve. However, our efforts did not cease at that point. The integration of openlittermap data resulted in a notable enhancement of our model's performance, hence improving its capacity to effectively distinguish between different types of trash. Moreover, the adaptability and effectiveness of the system were demonstrated by trials conducted using benchmark datasets, extending its applicability beyond the confines of our specific dataset. The conclusion of this chapter instills a feeling of fulfillment and a sense of eagerness for further investigation in the realm of trash classification using deep neural networks. In the subsequent chapter, an analysis of the findings will be conducted, followed by the proposal of potential directions for future research.

Chapter 5

Conclusion

The purpose of this research is to identify instances of littering in outdoor settings that are representative of the environment in Bangladesh. As far as we are aware, there are no benchmark datasets available for multiple garbage detection in Bangladesh, which would be helpful for the construction of any intelligent waste sorter. In order to accomplish this, we compiled a new dataset consisting of ten separate categories using images taken in Bangladesh's natural environments as the source material. We included some data from openlittermap to our existing dataset so that our model could better generalize its findings. Our extended dataset includes a total of 4418 images, with some images representing a single trash category while others representing numerous trash categories. The dataset is available for use in further research and may be acquired by contacting the authors of the study. The codes, while presently inaccessible, are being actively developed for inclusion in the repository along with the dataset in the coming period, with the aim of enhancing the reproducibility of our research. Many tests were carried out on our dataset with the assistance of the YOLOv5 models. We employed four distinct models that were all part of the YOLOv5 family. In the majority of the tests, the YOLOv5x model performed better than the other versions, despite the fact that it required more time to complete the inference. Our experiments involved using the YOLOv5x model to evaluate our dataset and an extended dataset. The mAP value for the extended dataset was higher than our original dataset. In addition to this, we used our models on two previously collected datasets, namely TACO and PlastOpol. With the TACO dataset, which includes sixty different classes, we got an mAP of 25.5%. Furthermore, we conducted single-class detection experiments on all our datasets, assuming all classes belong to the trash category. Our single-class detection experiments achieved better results than the current state-of-the-art methods for TACO and PlastOpol datasets, with an mAP of 55% for TACO (YOLOv5s) and 88.4% for PlastOpol. This indicates the effectiveness of our approach in detecting trash in outdoor environments. The mAP for a single class category in our expanded dataset was found to be 84.4%.

5.1 Future Works

As we draw this thesis to a close, we wish to offer some concise insights regarding the constraints of our study and avenues for future expansion.

- Expanding the dataset to include a wider array of categories necessitates a substantial effort in terms of data collection. This involves not only finding sources for additional categories but also ensuring that the data is representative and of sufficient quality. The challenge lies in the diversity of data sources, formats, and the need to curate and preprocess data for consistency.
- The issue of class imbalance is exacerbated when incorporating a wider array of categories. Some categories may have significantly fewer instances, making it challenging to train and evaluate machine learning models effectively. This imbalance can lead to model biases and reduced performance for underrepresented categories.

We aim to incorporate a broader range of categories, such as ocean garbage, to achieve a more evenly distributed dataset. Other object identification methods, such as SSD, Mask R-CNN, and EfficientDet, will be tested in the future using our dataset. We will investigate the advancement of custom network architecture and conduct experiments involving different combinations of hyperparameters in order to better optimize prediction accuracy. Our current studies did not include a disposable or recyclable tag on the labels. It will also be incorporated in the future.

List of Publications

During the course of this program, the following publications were published in relation to this research.

[1] Dhrubajyoti Das, Kaushik Deb, Taufique Sayeed, Pranab Kumar Dhar, and Tetsuya Shimamura, "Outdoor Trash Detection in Natural Environment Using a Deep Learning Model," in IEEE Access, vol. 11, pp. 97549-97566, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3313166.

[2] Dhrubajyoti Das, Anik Sen, Syed Md Minhaz Hossain, and Kaushik Deb, "Trash Image Classification Using Transfer Learning Based Deep Neural Network," In International Conference on Intelligent Computing & Optimization, pp. 561-571. Cham: Springer International Publishing, 2022, doi: https://doi.org/10.1007/978-3-031-19958-5_53.

Bibliography

[1] J. Gutberlet and S. M. N. Uddin, "Household waste and health risks affecting waste pickers and the environment in low-and middle-income countries," *International journal of occupational and environmental health*, vol. 23, no. 4, pp. 299–310, 2017.

[2] Y. Chartier, *Safe management of wastes from health-care activities*. World Health Organization, 2014, Accessed: 15.03.2023.

[3] *Asia-europe foundation (asef) - home*, Accessed: 15.03.2023, Oct. 2021. [Online]. Available: https://asef.org/wp-content/uploads/2021/11/ASEFSU23-Background-Paper_Waste-Management-in-Bangladesh.pdf.

[4] S. A. Urme, M. A. Radia, R. Alam, *et al.*, "Dhaka landfill waste practices: Addressing urban pollution and health hazards," *Buildings and Cities*, vol. 2, no. 1, pp. 700–716, 2021. DOI: 10.5334/bc.108.

[5] M. Ashikuzzaman and M. H. Howlader, "Sustainable solid waste management in bangladesh: Issues and challenges," *Sustainable waste management challenges in developing countries*, pp. 35–55, 2020.

[6] N. Ferronato and V. Torretta, "Waste mismanagement in developing countries: A review of global issues," *International journal of environmental research and public health*, vol. 16, no. 6, p. 1060, 2019.

[7] H. Zhou, X. Yu, A. Alhaskawi, *et al.*, "A deep learning approach for medical waste classification," *Scientific reports*, vol. 12, no. 1, p. 2159, 2022.

[8] C. Shi, C. Tan, T. Wang, and L. Wang, "A waste classification method based on a multilayer hybrid convolution neural network," *Applied Sciences*, vol. 11, no. 18, p. 8572, 2021.

[9] R. Sultana, R. D. Adams, Y. Yan, P. M. Yanik, and M. L. Tanaka, "Trash and recycled material identification using convolutional neural networks (cnn)," in *2020 SoutheastCon*, IEEE, 2020, pp. 1–8.

[10] N. Jayawickrama, R. Ojala, J. Pirhonen, K. Kivekäs, and K. Tammi, "Classification of trash and valuables with machine vision in shared cars," *Applied Sciences*, vol. 12, no. 11, p. 5695, 2022.

[11] K. R. Ahmed, "Smart pothole detection using deep learning based on dilated convolution," *Sensors*, vol. 21, no. 24, p. 8406, 2021.

[12] G. Thung and M. Yang, "Trashnet," *GitHub repository*, 2016. [Online]. Available: https://github.com/garythung/trashnet.

[13] P. F. Proença and P. Simoes, "Taco: Trash annotations in context for litter detection," *arXiv preprint arXiv:2003.06975*, 2020. [Online]. Available: https://arxiv.org/abs/2003.06975.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[16] 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.

[17] W. Liu, D. Anguelov, D. Erhan, *et al.*, "Ssd: Single shot multibox detector," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, Springer, 2016, pp. 21–37.

[18] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[20] S. Lynch, "Openlittermap. com–open data on plastic pollution with blockchain rewards (littercoin)," *Open Geospatial Data, Software and Standards*, vol. 3, no. 1, pp. 1–10, 2018.

[21] A. Masand, S. Chauhan, M. Jangid, R. Kumar, and S. Roy, "Scrapnet: An efficient approach to trash classification," *IEEE access*, vol. 9, pp. 130 947–130 958, 2021.

[22] Z. Yang, Y. Bao, Y. Liu, Q. Zhao, H. Zheng, and Y. Bao, "Research on deep learning garbage classification system based on fusion of image classification and object detection classification," *Mathematical Biosciences and Engineering*, vol. 20, no. 3, pp. 4741–4759, 2023.

[23] B. Fu, S. Li, J. Wei, Q. Li, Q. Wang, and J. Tu, "A novel intelligent garbage classification system based on deep learning and an embedded linux system," *IEEE Access*, vol. 9, pp. 131 134–131 146, 2021.

[24] A. H. Vo, M. T. Vo, T. Le, *et al.*, "A novel framework for trash classification using deep transfer learning," *IEEE Access*, vol. 7, pp. 178 631–178 639, 2019.

[25] M. Córdova, A. Pinto, C. C. Hellevik, *et al.*, "Litter detection with deep learning: A comparative study," *Sensors*, vol. 22, no. 2, p. 548, 2022.

[26] C. Liu, N. Xie, X. Yang, *et al.*, "A domestic trash detection model based on improved yolox," *Sensors*, vol. 22, no. 18, p. 6974, 2022.

[27] C. Wu, Y. Sun, T. Wang, and Y. Liu, "Underwater trash detection algorithm based on improved yolov5s," *Journal of Real-Time Image Processing*, vol. 19, no. 5, pp. 911–920, 2022.

[28] M. Tian, X. Li, S. Kong, L. Wu, and J. Yu, "A modified yolov4 detection method for a vision-based underwater garbage cleaning robot," *Frontiers of Information Technology & Electronic Engineering*, vol. 23, no. 8, pp. 1217–1228, 2022.

[29] D. O. Melinte, A.-M. Travediu, and D. N. Dumitriu, "Deep convolutional neural networks object detector for real-time waste identification," *Applied Sciences*, vol. 10, no. 20, p. 7301, 2020.

[30] P. Kaur, B. S. Khehra, and E. B. S. Mavi, "Data augmentation for object detection: A review," in *2021 IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*, IEEE, 2021, pp. 537–543.

[31] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, 2020, ISSN: 2078-2489. DOI: 10.3390/info11020125. [Online]. Available: https://www.mdpi.com/2078-2489/11/2/125.

[32] G. Jocher, A. Chaurasia, A. Stoken, *et al.*, *ultralytics/yolov5: v6.1 - TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference*, version v6.1, Feb. 2022. DOI: 10.5281/zenodo.6222936. [Online]. Available: https://doi.org/10.5281/zenodo.6222936.

[33] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018. [Online]. Available: https://arxiv.org/abs/1804.02767.

[34] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934.

[35] B. Mahaur and K. Mishra, "Small-object detection based on yolov5 in autonomous driving systems," *Pattern Recognition Letters*, vol. 168, pp. 115–122, 2023.

[36] W. Jia, S. Xu, Z. Liang, *et al.*, "Real-time automatic helmet detection of motorcyclists in urban traffic using improved yolov5 detector," *IET Image Processing*, vol. 15, no. 14, pp. 3623–3637, 2021.

[37] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Csp-net: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 390–391.

[38] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8759–8768.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[40] T.-Y. Lin, M. Maire, S. Belongie, *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, Springer, 2014, pp. 740–755.

[41] N. V. Kumsetty, A. B. Nekkare, S. Kamath, *et al.*, "Trashbox: Trash detection and classification using quantum transfer learning," in *2022 31st Conference of Open Innovations Association (FRUCT)*, IEEE, 2022, pp. 125–130.

[42] S. N. Endah, I. N. Shiddiq, *et al.*, "Xception architecture transfer learning for garbage classification," in *2020 4th International Conference on Informatics and Computational Sciences (ICICoS)*, IEEE, 2020, pp. 1–4.

[43] S. Niu, J. Wang, Y. Liu, and H. Song, "Transfer learning based data-efficient machine learning enabled classification," in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, IEEE, 2020, pp. 620–626.

[44] D. Das, A. Sen, S. M. M. Hossain, and K. Deb, "Trash image classification using transfer learning based deep neural network," in *Intelligent Computing & Optimization: Proceedings of the 5th International Conference on Intelligent Computing and Optimization 2022 (ICO2022)*, Springer, 2022, pp. 561–571.

[45] H. Panwar, P. Gupta, M. K. Siddiqui, *et al.*, "Aquavision: Automating the detection of waste in water bodies using deep transfer learning," *Case Studies in Chemical and Environmental Engineering*, vol. 2, p. 100 026, 2020.

[46] S. Neelakandan, M. Prakash, B. Geetha, *et al.*, "Metaheuristics with deep transfer learning enabled detection and classification model for industrial waste management," *Chemosphere*, vol. 308, p. 136 046, 2022.

[47] T. NgoGia, Y. Li, D. Jin, J. Guo, J. Li, and Q. Tang, "Real-time sea cucumber detection based on yolov4-tiny and transfer learning using data augmentation," in *Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II 12*, Springer, 2021, pp. 119–128.

[48] Y. Chen, J. Sun, S. Bi, C. Meng, and F. Guo, "Multi-objective solid waste classification and identification model based on transfer learning method," *Journal of Material Cycles and Waste Management*, vol. 23, no. 6, pp. 2179–2191, 2021.

[49] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri, "Activation functions in deep learning: A comprehensive survey and benchmark," *Neurocomputing*, 2022.

[50] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Networks*, vol. 107, pp. 3–11, 2018.

[51] Q. Xu, Z. Zhu, H. Ge, Z. Zhang, and X. Zang, "Effective face detector based on yolov5 and superresolution reconstruction," *Computational and Mathematical Methods in Medicine*, vol. 2021, pp. 1–9, 2021.

[52] R. Padilla, S. L. Netto, and E. A. Da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 international conference on systems, signals and image processing (IWSSIP)*, IEEE, 2020, pp. 237–242.

[53] C. Li, L. Li, H. Jiang, *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022. [Online]. Available: https://arxiv.org/abs/2209.02976.

[54] G. Jocher, A. Chaurasia, and J. Qiu, *YOLO by Ultralytics*, version 8.0.0, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics.

[55] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016. [Online]. Available: https://arxiv.org/abs/1606.08415.

[56] S. Majchrowska, A. Mikołajczyk, M. Ferlin, *et al.*, "Deep learning-based waste detection in natural and urban environments," *Waste Management*, vol. 138, pp. 274–284, 2022.

[57] I. Borowy, "Medical waste: The dark side of healthcare," *História, Ciências, Saúde-Manguinhos*, vol. 27, no. suppl 1, pp. 231–251, 2020. DOI: 10.1590/s0104-59702020000300012.