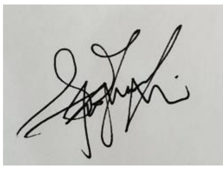
APPLICATION DEVELOPMENT FOR PLASTIC BOTTLE DETECTION USING DEEP LEARNING

FONG YUN XIN

UNIVERSITI TUNKU ABDUL RAHMAN

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : 

Name : Fong Yun Xin

ID No. : 2002937

Date : 25/04/2024

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"APPLICATION DEVELOPMENT FOR PLASTIC BOTTLE DETECTION USING DEEP LEARNING"** was prepared by **FONG YUN XIN** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Software Engineering with Honours at Universiti Tunku Abdul Rahman.

Approved by,

Signature	:	kckhor
Supervisor	:	Khor Kok Chin
Date	:	11/5/2024

Signature	:	
Co-Supervisor	:	
Date	:	

ACKNOWLEDGEMENTS

My greatest thanks go to my supervisor, Dr. Khor Kok Chin, who has supported me throughout my final year project and guided me patiently from beginning to the end of the project. Moreover, I am also thankful to my co-supervisor, Dr. Ting Cheng Hunt for helping in this project as well.

Next, I want to sincerely thank my family member for providing emotional support during my entire final year project. Without their support, I would not be able to complete the project.

Last but not least, I would like to thanks to all my friends who also help me a lot during my journey in completing my final year project. Special thanks to all the contributors for YOLOv8 and the author of YOLOv8, Ultralytics for providing detailed documentation that helps me a lot to complete my project.

ABSTRACT

Nowadays, recycling centers still rely on human workers which is low efficiency and working environment is bad for the human workers. Hence, deep learning is introduced to detect the plastic bottles on the moving conveyer belt in the recycling centers. In this project, three pre-trained deep learning models is selected to train and detect the plastic bottles. The three selected pre-trained deep learning models are YOLOv8, Faster R-CNN and SSD. The results show that YOLOv8 achieved the highest mean average precision for the custom dataset which is 0.923 compared to Faster RCNN and SSD. Thus, YOLOv8 is selected and further tested with the real video from the recycling center to detect the plastic bottles on the conveyer belt. In the video, YOLOv8 achieved an average precision of 0.3026 in detecting the plastic bottles, but the average precision significantly improved to 0.6783 when the waste products is less overlapping on the moving conveyer belt. The application had passed the user satisfactory survey and user acceptance test, so it is easy to be used for people who does not have knowledge in deep learning.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

mAP	Mean Average Precision
FPS	Frame Per Second
IOU	Intersection Over Union
NMS	Non-Max Suppression
FPN	Feature Pyramid Network
CNN	Convolutional Neural Network
SPP	Spatial Pyramid Pool
PET	Polyethylene terephthalate

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Human misuse of plastics has caused serious environmental pollution ever since plastics were first invented. The environment is now endangered due to plastic pollution. Improper management of plastic waste will have a negative impact on ecosystems and public health especially in developing countries which do not take plastic pollution issue seriously. Malaysia was one of the biggest importers of plastic trash in 2021. Malaysia brought in over 500,000 tons while only exporting 11,000 tons (Statista Research Department, 2023), which made the situation worse. Malaysia needs to deal with a significant issue of plastic waste that is harming both terrestrial and marine environments. (Chen et al., 2021) Since poor management of plastic waste would have significant detrimental consequences on both human health and the environment, Malaysia must focus on controlling plastic waste to minimize the plastic pollution to the environment.

Nowadays, people are giving more attention to environmental preservation as awareness of protecting the environment had grown among the general public. Thus, people have started to protect the environment by recycling existing plastics and minimising the production of more plastics to protect the environment. Recycle becomes the solution for reducing the plastic waste in the environment. Plastic must go through a number of steps in order to be recycled, including gathering, sorting, and reprocessing the plastic so that it can be utilized to make new products. Even though advanced recycling equipment has made the process easier for human workers, it cannot replace human workers in plastic waste sorting. Human workers still play an important role in the sorting of plastic waste. Manual sorting is still required to separate the plastic waste and categorise them to further process the plastic waste. However, manual sorting leads a few problems in term of efficiency, effectiveness, and safety hazards. The problem faced must be solved to improve the recycling process to be much more efficient, effective and ensure the safety of human workers.

1.2 Importance of the Study

To replace or reduce human involvement in sorting plastic waste, the application development of plastic bottle detection using deep learning holds great potential to address this challenge. In this project, the focus is on the sorting process for the recycling process for the PET-plastic bottle wastes and the detection and classification of other plastic waste will only be considered in future. By developing a plastic bottles detection application, the application is expected to optimise and streamline recycling operations in the sorting process, thereby improving efficiency and effectiveness.

The application will leverage advanced deep learning models to accurately identify and classify PET-plastic bottles based on characteristics such as bottle shape, size, colour, labels, and surface texture. The plastic bottle detection application is a possible solution for solving these problems as there are pre-trained deep learning models exist that can be implemented in the plastic bottle detection application to detect PET-plastic bottles with high accuracy. After training the pre-trained deep learning models, the deep learning model that has the best performance among the other pre-trained deep learning models in detecting PET-plastic bottles can be implemented in real-life situations to replace human workers in sorting the plastic bottles waste.

1.3 Problem Statement

The current problem in the recycling industry for sorting plastic bottle waste is ineffective in sorting out plastic bottle waste. Human error is a common issue in manual sorting. The error can occur due to fatigue, distractions, or differences in perception and judgment. Not just the sorting process will be prolonged after long working hours, but human error may also increase as time goes on. Incorrect sorting can lead to recycling contamination, where plastic bottles end up in the wrong category and cannot be properly recycled. Besides, given that Malaysia's plastic recycling predominantly depends on manual sorting, the task of categorizing plastic bottles falls upon the public or recycling factory workers themselves. However, due to insufficient knowledge about proper plastic sorting, the sorting process faces obstacles, thus presenting a significant challenge for the plastic recycling industry (Tan et al., 2022).

Furthermore, there is also a problem for normal people using a deep learning model without using a user interface (UI). Deep learning models typically require complex programming and command-line interactions, making them challenging for people who do not have a technical background to utilise the deep learning tools. As the targeted user for the deep learning model for plastic bottle detection will be the workers in recycling factories, it is assumed that they do not have knowledge of deep learning. Thus, it will be a challenge for the workers to use the deep learning model for the plastic bottles sorting process without user interface.

1.4 Aim and Objectives

The aim of application development for plastic bottle detection using deep learning is to solve these problems by automating the sorting process. Deep learning is used in the development of the plastic bottle detection application to enable it to recognize and classify plastic bottles according to their characteristics. The plastic bottle detection application that is developed using deep learning models is expected to differentiate PET-plastic bottles from other wastes based on the PET-plastic bottles' characteristics, such as shape, colour, or texture. The application aims to achieve the following objectives:

1. To implement and evaluate three pre-trained models for PET-plastic bottle detection and find the best performer.

The plastic bottle detection application should be able to accurately identify, and sort plastic bottles based on their specific characteristics. This can be achieved by training the model with a dataset containing PET-plastic bottles. By doing this, the sorting process takes less time and ensures the PET-plastic bottles are placed into the correct recycling categories and enhances the overall effectiveness of the PET-plastic bottle sorting process.

2. To develop an application for plastic bottle detection using the best performer and evaluate the usability of the application.

As the application's target users will be the workers in the recycling factory, it is assumed that the workers do not know about deep learning. It will be challenging for workers without deep learning knowledge to use the deep learning model. Therefore, a user interface is needed to assist users who lack deep learning knowledge to interact with the pre-trained deep learning model without any issues. For example, users able to open the webcam in the application and the application should return whether any plastic bottle is identified based on the webcam input.

1.5 Project Solution

The above-mentioned problems can be solved by using the existing pre-trained deep learning model. Faster R-CNN, SSD and YOLO (You Only Look Once) are some of the popular deep learning models. In this project, the deep learning model with the best performance will be utilized to distinguish PET-plastic bottles from other types of waste. Among all the pre-trained deep learning models, YOLO is one of the most popular pre-trained deep learning models in object detection due to its remarkable balance of speed and accuracy, making it suitable for real time detection. However, Faster R-CNN, SSD and YOLO will also be trained and evaluate their ability in detecting PET-plastic bottles.

Once the pre-trained deep learning model with the best performance has been selected, an application will be developed to enable workers in the recycling factory to monitor the sorting process of plastic bottles. The application will allow workers to easily monitor the plastic bottle sorting process without requiring extensive knowledge of deep learning. Users can open the webcam, and the application will highlight any detected plastic bottles by drawing boxes around them within the images. This simplifies the complex deep learning tasks for individuals who are not familiar with the deep learning.

1.6 Project Approach

The methodology used in the project is Kanban. There are a few reasons why Kanban is chosen to use in the project. In brief, Kanban provides a simple and flexible project management framework that fits the needs of a solo developer and the project's workflow perfectly. Kanban methodology is chosen for this project because of its flexibility in work prioritization, capability in visualizing the project progress, and promotion of continuous development.

Firstly, Kanban facilitates task organization and prioritization for quick project development. The columns "To Do," "In Progress," and "Done" are used to represent the various project stages. The developer can maintain organization and attention while monitoring activities and project progress because of this overview provided by the Kanban board. Additionally, Kanban enables the developer to carry out other tasks back and forth without interfering with the training of deep learning models or the development of applications throughout the project. This adaptability ensures the project stays on schedule and maintains steady progress.

Kanban also emphasizes continuous delivery, allowing the developer to carry out small updates and enhancements to the application or pre-trained deep learning model. This iterative process encourages quick feedback and frequent improvements. Kanban encourages a culture of continuous improvement by encouraging regular progress reviews, assessing the performance of the deep learning models, and improving the application. Last but not least, the Kanban technique reduces the need for lengthy planning, allowing developers to focus on building the application and training the deep learning models.

1.7 Scope and Limitation of the Study

The project aims to develop an application for plastic bottle detection using a deep learning model. The pre-trained deep learning model chosen for comparison in this project is Faster R-CNN, SSD and YOLO Object Detection. Different deep learning models will be selected for comparison, and the pre-trained deep learning model with the best performance will be implemented in the plastic bottle detection application. The application should be able to distinguish plastic bottles from other waste and sort the identified plastic bottles, allowing the recycling factory to sort and send them for further recycling.

Several tasks need to be completed in this project. Firstly, a diverse dataset of PET-plastic bottle images must be prepared. The dataset of the plastic bottle images will include PET-plastic bottles in various shapes, colours, and sizes (500mililitres to 1000mililitres). Sufficient images should be included in the dataset to avoid underfitting or overfitting problems in the training of the deep learning models. The dataset will be divided into three categories: training set, validation set, and testing set. Secondly, research will be conducted to select a few suitable pre-trained deep-learning models to train models. The deep learning model will be trained using the prepared dataset. The dataset will export according to the format required by the deep learning model for training the selected model in order to develop the plastic bottle detection application. More than one pre-trained model will be chosen for training to compare the performance of the selected pre-trained models in detecting plastic bottles.

Thirdly, after pre-training the deep learning models with the training dataset, the detection accuracy of the chosen pre-trained models for plastic bottles will be verified using the validation dataset. This step involves adjusting the pre-trained deep learning models' parameters, such as batch sizes and number of epochs, to optimise the pre-trained deep learning models' performance in detecting plastic bottles. The pre-trained deep learning models are expected to have a better performance after adjusting the parameters.

Fourthly, the performance of the pre-trained deep learning models will be evaluated by conducting tests using testing datasets and real scenarios data. Performance metrics such as mean average precision will be measured to assess the pre-trained deep learning model's performance in accurately and effectiveness of detecting plastic bottles. The results of the pre-trained deep learning models will be compared, and the pre-trained deep learning model with the highest accuracy will be chosen for the development of the application for plastic bottle detection.

After the deep learning models is trained and ready to be used, the project will need to start developing an application that can let the workers in the recycling factory interact with the pre-trained deep learning model. The application should allow users to open webcam to detect plastic bottles. The application will return feedback to users by bounding the location of the detected plastic bottles if any plastic bottles are found in the input of the webcam.

Lastly, the entire application development process will be documented, including the choice of the pre-trained deep learning model, development, and implementation details of the application. Additionally, potential areas for future improvements will be identified, such as optimising the pre-trained deep learning model for speed or extending the pre-trained deep learning model to detect other recyclable items.

1.7.1 User Scope

This project is targeted at workers in recycling factories who are in charge of the waste sorting process in recycling factories.

1.7.2 System Scope

The application should be able to detect and differentiate PET-plastic bottles from others waste. The plastic bottle detection application utilises the existing pre-trained deep learning model to study the characteristics of PET-plastic bottles and sort the PET-plastic bottles from other wastes.

1.7.3 The Limitation of the Study

The application for this project will be developed using Python programming language. Besides, the application will only focus on detecting PET-plastic bottles waste but not any other wastes (Other waste will not be considered in this project). Moreover, the size of the PET-plastic bottles is ranging from 500 millilitres to 1000 millilitres only.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter examines various aspects, including distinguishing between machine learning and deep learning, tracing the evolution of object detection, comparing one-stage detectors with two-stage detectors, exploring evaluation metrics and Kanban Development Methodology will be discussed thoroughly. Additionally, it reviews recent research in related fields and provides comparisons. The primary goals of this chapter are:

a) To gain a comprehensive understanding of deep learning and object detection concepts.
b) To acquire knowledge on the selection of deep learning models and techniques for improvement.
c) To get suitable evaluation metric to study the performance of the deep learning models.

2.2 Machine Learning and Deep Learning

Machine learning is a subset of both artificial intelligence (AI) and computer science that involves the use of data and algorithms to imitate human learning patterns and improve its accuracy progressively. Deep learning refers to a neural network that is built by at least three or more layers and it is a subset of machine learning. The public is often confused by the difference of the terms between artificial intelligence, machine learning, and deep learning. However, it is crucial to comprehend that they are nested within one another in order to understand their relationship. Machine learning and deep learning both fall under the category of machine learning, which is a subset of artificial intelligence and has the broadest applications.

Moreover, the primary distinction between machine learning and deep learning lies in how each algorithm learns and how much data they utilize. Deep learning streamlines the feature extraction process, reducing the need for manual human intervention. Deep learning excels with large datasets and the potential of deep learning capability is captivating as the deep learning able to explore of unstructured data for training, especially considering that most of the data existed are in unstructured formats. For example, the cat and dog recognition is an example of unstructured data. By observing patterns in data, deep learning models can effectively categorize the inputs into groups based on similarities or differences in the images. In brief, deep learning models require more data points to enhance accuracy, while machine learning models perform well with less data due to their underlying data structure (IBM Data and AI Team, 2023). Hence, the reason why deep learning is chosen over traditional machine learning for object detection for the project due to its ability to automatically learn and extract complex features the data, thus no human intervention is needed for feature extraction. Furthermore, the availability of state-of-the-art models like Faster R-CNN, SSD and YOLO models that offer top-tier accuracy and speed, making them ideal for the plastic bottle detection application.

2.3 Evolution of Object Detection.

Over the past two decades, object detection has undergone significant improvements. The introduction of deep learning has brought a revolutionary effect to object detection. Some examples of traditional object detection algorithms before the introduction of deep learning include the Viola-Jones Detector, HOG-Detector, and DPM. After the integration of deep learning into object detection, numerous state-of-the-art algorithms have emerged. These object detection algorithms can be categorized into two major groups: one-stage detectors and two-stage detectors. RCNN, Fast RCNN, Faster RCNN, Mask R-CNN, SPPNet, Pyramid Networks/FPN, and G-RCNN are some of the examples of two-stage object detection algorithms while YOLO, YOLOv3, YOLOv5 and SSD are the example of one-stage object detection algorithms (Boesch, 2023). The revolution of the object detection algorithm after the introduction of deep learning makes the object detection process to be more effective and efficient than ever.

From the revolution of object detection, it is known that object detection can be carried out through either traditional detection methods or deep learning models. Traditional detection methods which are the image processing techniques are typically unsupervised and have the advantage of not requiring annotated images. However, they are limited when it comes to complex scenarios without a uniform background, instances of occlusion (partially hidden objects), issues with illumination and shadows, and cluttered scenes. On the other hand, object detection utilising deep learning models, whether supervised or unsupervised, is the norm in computer vision tasks. Deep learning-based object detection excels in handling challenges like occlusion, complex scenes, and difficult lighting conditions. However, they demand substantial training data, making the image annotation process labour-intensive and costly. For example, labelling 500,000 images to train a custom deep learning object detection algorithm would be a time-consuming task. However, 500,000 is still considered a small dataset for training deep learning models. Nevertheless, many benchmark datasets, such as MS COCO, Caltech, KITTI, Pascal VOC, and V5, offer labelled data to facilitate the development of object detection deep learning models for the researchers.

In this project, deep learning is chosen for the plastic bottle detection because traditional detection methods face difficulty in selecting which feature is important and needed to be extracted. The decision regarding which feature to extract relies on the judgement of the computer vision engineer. This determination often necessitates a lengthy and iterative trial-and-error procedure to find out the most suitable features for effectively characterizing diverse object classes. By using a deep learning model that is trained by the provided dataset, the neural networks can identify the patterns within various categories of images and autonomously determine the most informative and prominent features for each object class (O’Mahony et al., 2020). Thus, it is widely acknowledged that deep learning consistently outperforms traditional detection methods, although they do come with certain trade-offs in terms of computational demands. Hence, deep learning that can do end-to-end learning using Convolutional Neural Networks becomes a good solution for object detection.

2.3.1 One-stage detectors and Two-stage detectors

One-stage detectors and two-stage detectors are the two primary categories of object detection algorithms that leverage deep learning. Each of the object detection algorithm has its own advantages and disadvantages. For example, one-stage detectors prioritized high inference speed, making them suited for real-time applications. On the other hand, the localization and recognition accuracy of two-stage detectors is superior to one-stage detectors. Two-stage detectors consist of two phases to distinguish between these two methodologies. The Region Proposal Network (RPN), the initial stage, forecasts potential boundary boxes. The second stage is object classification using bounding-box regression and features taken from the suggested regions. Some of the examples for two-stage detectors include RCNN, Fast RCNN, and Faster RCNN. In contrast, one-stage detectors accomplish bounding box prediction in a single step, eliminating the need for a separate region proposal stage. They utilize a grid box and anchor mechanisms to both locate regions of interest within an image and define the shape parameters of objects. YOLO and SSD are a notable example of a one-stage detector.

Two-stage detectors typically achieve high accuracy rates but are slower than one-stage detectors, which are generally faster at object detection but may have lower accuracy. In essence, there is a trade-off between accuracy and speed in object detection, and the choice of a deep learning model depends on the specific application. For the application of plastic bottle detection, one-stage detectors are more suitable due to their exceptional speed in real-time object detection. Although one-stage detectors may not offer accuracy levels as high as two-stage detectors, their accuracy is still adequate for real-time use. Furthermore, recent advancements in one-stage detectors have significantly improved their detection accuracy. In this project, one-stage detectors, specifically YOLO models, are selected to investigate their performance in detecting plastic bottles. Additionally, the project aims to explore enhancements to the YOLO architecture to further improve detection accuracy, as discussed in the following section.

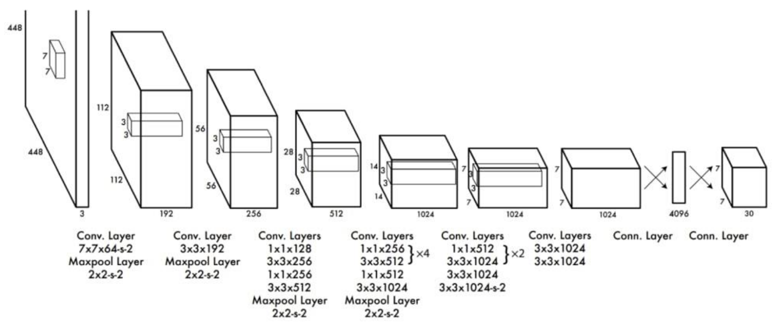
2.4 YOLO

One example of a one-stage detector is a deep learning model called You Only Look Once (YOLO). By incorporating convolutional neural network (CNN) that simultaneously predicts bounding boxes and class probabilities, the You Only Look Once (YOLO) deep learning model revolutionized object detection. This was different from earlier object detection techniques that used classifiers as detectors. With a significant performance advantage over competing algorithms, this novel strategy propelled YOLO to the top of the real-time object detection field. YOLO completes all predictions in a single step through a single fully connected layer, as opposed to techniques like Faster RCNN that require a two-step process of region proposal and recognition.

2.4.1 YOLO Architecture

Over the years, YOLO has developed in various versions, but they are mainly enhancements of the original model. The original YOLO model is crucial for understanding how YOLO works, as it serves as the foundation. So, studying the architecture of the first YOLO model remains important for grasping its fundamental principles and mechanisms. The convolutional neural network is used by YOLO to begin by taking a picture as input and detecting items within it. The YOLO model's first 20 convolutional layers are pre-trained on ImageNet. Then, using a method proven to boost performance, convolutional and fully connected layers are added to this pre-trained model to adjust it for object detection. Both class probabilities and bounding box coordinates are predicted by the last fully connected layer in the YOLO architecture. YOLO makes prediction on the input image by dividing the input image into a SxS grid and each of the grid cell is responsible for detecting an object if the centre of an object lies within that grid cell. Then, b bounding boxes are predicted in each grid cell, and confidence scores are given to those boxes. The output images will consist of bounding boxes to show the location of the object, with each bounding box associated with a confidence score indicating how confident the model believes the prediction is correct (Kondu, 2023).

To ensure that each object in the images is only associated with one bounding box predictor, YOLO uses a single predictor that returns the highest intersection over union (IOU) with the ground truth bounding box. This specialization among bounding box predictors enhances the model's overall recall score because each predictor becomes more capable at predicting specific object sizes, aspect ratios, or classes. Another important technique used in YOLO models is called non-maximum suppression (NMS). NMS is a post-processing step that enhances the accuracy and efficiency in object detection. In object detection, multiple bounding boxes can be generated for a single object, leading to overlap or variations in their positions.
NMS identifies and eliminates redundant or inaccurate bounding boxes by using an intersection over union (IOU) threshold to ensure that only one bounding box per object is retained in the final output.
For example, if the IOU threshold is set to 0.5, it will remove the bounding box that has a lower confidence score if the IOU of the bounding box is higher than 0.5, because the two bounding boxes have a high chance of referring to the same detected object (Kondu, 2023). Thus, the following section will provide a detailed explanation of the newer versions of YOLO after understanding the architecture of YOLO.



The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

Figure 2.1: YOLO architecture (Redmon et al., 2016)

2.4.2 YOLOv2

In 2016, an upgraded version of YOLO, known as YOLO v2 or YOLO9000, was introduced to enhance the original YOLO algorithm. One of the improvement in YOLO v2 is the implementation of batch normalization, which enhances model accuracy and stability. YOLO v2 also employs a different Convolutional Neural Network (CNN) structure called Darknet-19. Darknet-19 is a modified form of the VGGNet architecture featuring simplified progressive convolution and pooling layers to achieve higher speed and accuracy while expanding its ability to detect a broader range of object classes. Additionally, YOLO v2 employs a multi-scale training approach that randomly chooses a new image dimension size for every ten batches. This technique improves the ability of detection for small objects. Additionally, YOLO v2 features a new loss function that is based on the class probabilities and the sum of squared errors between the predicted and actual bounding boxes. The loss function is created especially for object detection tasks. (Kondu, 2023).

2.4.3 YOLOv3

YOLO v3 was introduced in 2018 as a step forward from YOLO v2. In YOLO v3, one of the significant advancements of YOLOv3 is the adoption of a new CNN architecture termed Darknet-53 with 53 convolutional layers. The newly adopted CNN architecture not just more powerful than Darknet-19, but also more efficient than ResNet-101 by 1.5 times faster and ResNet-152 by 2 times faster. Moreover, YOLO v3 stepped up from YOLOv2 by incorporating anchor boxes of different scales and ratios to achieve better matching for the sizes and shapes of identified objects. Furthermore, YOLO v3 introduces feature pyramid networks (FPN) which are a type of CNN architecture used for predicting objects at multiple scales. FPN creates a stack of feature maps to form a pyramid structure and in each level of this feature pyramid network, it is used to spot objects with different scale. This innovation greatly boosts the detection performance, especially for objects in small size, as the model can perceive objects from multiple scales. Alongside these advancements, YOLO v3 excels in handling a wider range of object sizes and shapes (Kondu, 2023).

2.4.4 YOLOv4

For the main improvment in YOLOv4 compared to YOLOv3, YOLOv4 employs a new Convolutional Neural Network (CNN) design named CSPNet (Cross Stage Partial Network). CSPNet is a modified version of the ResNet architecture, that is custom-made for object detection. Despite having a relatively modest structure consisting of just 54 convolutional layers, it delivers outstanding performance in numerous object detection evaluations. Furthermore, another improvement for YOLOv4 is introduced called k-means clustering for creating anchor boxes. This method is used alongside with anchor boxes that have various sizes and aspect ratios to better correspond with the dimensions and shapes of the objects being identified. Although YOLO v3 and v4 both use the same loss function to train the model, v4 adds a component known as "GHM loss." This component, which is a variant of the focal loss function, is intended to improve the model's performance, particularly when working with datasets that feature classes that are not uniformly distributed. Furthermore, YOLO v4 enhances the architecture of the Feature Pyramid Networks (FPNs) that were originally introduced in YOLO v3 (Kondu, 2023).

2.4.5 YOLOv5

In 2020, YOLO v5 was introduced by Ultralytics. YOLOv5 adopts a more complex architecture called EfficientDet, which is based on the EfficientNet network architecture. This advanced architecture in YOLO v5 enhances its accuracy and extends its ability to recognize a wider array of object categories. Furthermore, YOLO v5 differs from its predecessor in terms of the training data. The D5 dataset, which includes a total of 600 item categories, is used to train YOLO v5, whereas the original YOLO was trained using the 20-object category PASCAL VOC dataset. A novel technique for producing anchor boxes is used in YOLO v5 and is referred to as dynamic anchor boxes.

This approach employs a clustering algorithm to group the ground truth bounding boxes into clusters and then utilizes the cluster centroids as anchor boxes. This approach ensures that anchor boxes closely match the sizes and shapes of detected objects to give better prediction.

Moreover, the introduction of spatial pyramid pooling (SPP) is another big improvement in YOLO v5. SPP is a type of pooling layer that reduces the spatial resolution of feature maps. It enhances object detection performance, especially for small objects, by allowing the model to recognize objects at various scales. YOLO v5 makes improvements to the SPP architecture compared to YOLOv4, resulting in better outcomes for prediction. In terms of the loss function used for training, YOLO v4 and YOLO v5 are using similar loss function for training. However, YOLO v5 introduces a new component called "CIoU loss," which is an adaptation of the IoU loss function designed to enhance the model's performance, especially when dealing with datasets that have unevenly distributed classes (Kondu, 2023).

2.4.6 YOLOv6

In YOLOv6, EfficientRep is introduced as an efficient re-parameterizable backbone for deep learning models. In the case of smaller models during training, it employs RepBlock components, but during inference, it transforms RepBlocks into 3x3 convolutional layers (RepConv) with ReLU activation, optimizing computational efficiency. This approach capitalizes on the effectiveness of 3x3 convolutions on mainstream hardware. For larger models, an optimized CSPStackRep Block is employed, consisting of 1x1 convolution layers and sub-blocks with RepVGG blocks or RepConv, along with a CSP connection to balance accuracy and speed efficiently. This design ensures the backbone's scalability while maintaining computational efficiency and improving feature representation. These networks vary in scale, optimizing the balance between speed and accuracy; smaller models employ a single-path backbone, while larger models feature efficient multi-branch blocks. Furthermore, advanced detection techniques, including label assignment, loss functions, and data augmentation, are comprehensively explored and selectively adopted to improve performance (Li et al., 2022).

2.4.7 YOLOv7

The implementation of anchor boxes is the main focus of YOLOv7's upgrades. Anchor boxes are made up of pre-defined boxes with different aspect ratios that are used to identify various shapes of things. In comparison to earlier versions, YOLO v7 uses nine anchor boxes, which allows it to detect a wider variety of item shapes and sizes and lowers the likelihood of false positives. The introduction of a brand-new loss function known as "focal loss" in YOLOv7 is another significant advancement. The cross-entropy loss function used in prior versions of YOLO is less successful at recognizing small objects. Focal loss overcomes this drawback by placing less attention on instances with clear classifications and more importance on tough situations for difficult-to-detect objects. Additionally, YOLO v7 works at a greater resolution than YOLO v3 by processing photos at a 608 by 608 pixel size rather than the older version's 416 by 416 resolution. This improved resolution enables YOLO v7 to recognize smaller objects and increase accuracy in general. (Kondu, 2023).

2.4.8 YOLOv8

The improvement of YOLOv8 is it uses mosaic data augmentation to combine images to provide more context to the model to improve its performance. Other improvements for Yolov8 include its anchor-free detection, making it more generalized and able to speed up the learning rate for non-max suppression. The model backbone for Yolov8 has changed from C3 to C2f, as C2f will link all the outputs of the bottleneck module instead of just taking the last output of the bottleneck module. This will shorten the training process and improve gradient flow. Yolov8 also uses a decoupled head so that classification and regression are done separately, which will improve the model's performance. This separation is then combined into a single loss value and helps in network optimization, improving the detection localization accuracy and the classification accuracy (Boesch, 2024).

2.5 Single Shot MultiBox Detector (SSD)

SSD, also known as Single Shot MultiBox Detector is one of the examples of one stage detector. SSD does not rely on a separate network for region proposals like other two stage detectors. Instead, it explicitly predicts object boundary boxes and the classes that correspond to them in a single network run. In order to improve its accuracy, SSD introduces a number of key concepts: it makes use of multi-scale feature maps to detect objects of various sizes, small convolutional filters to predict object classes and adjustments for default boundary boxes, and separate filters for default boxes to handle variations in shape. The ability of SSD to be trained from beginning to end, which increases total accuracy, is a big advantage. SSD produces more predictions and offers better coverage for objects of various locations, sizes, and shapes. Impressively, SSD maintains comparable accuracy even with input photos with a lesser resolution that are just 300x300 pixels. SSD achieves real-time processing speed while outperforming cutting-edge models like Faster R-CNN in terms of accuracy by doing away with the necessity for a dedicated region proposal network and using lower-resolution images. (Hui, 2018).

2.6 Region-Based Convolutional Neural Network (R-CNN)

R-CNN were introduced by the UC Berkeley researchers in 2014. R-CNN is capable for recognizing 80 different types of objects in images. What made R-CNN different from traditional object detection methods was its use of a deep learning technique called a convolutional neural network (CNN) to identify features in images. R-CNN's structure closely resembled the usual object detection process, but with the key change of using CNN-based features. R-CNN had three main parts: it first created 2,000 region proposals using the Selective Search method; then it resized these proposals and extracted 4,096-length feature vectors from each one; finally, it used a pre-trained SVM algorithm to decide if each proposal represented the background or one of the object categories. In summary, R-CNN is a two-stage detector that transformed object detection by incorporating CNN-based feature extraction into its approach (Gad, 2020).

2.6.1 Faster RCNN

Faster R-CNN is the improvement version of the R-CNN, it delivers improved speed and efficiency in object detection, incorporates a Region Proposal Network (RPN) to enable end-to-end training for the entire object detection network. It introduces the RPN, a fully convolutional network that generates object proposals across various scales and aspect ratios. Essentially, the RPN acts as a guide for Fast R-CNN, telling it where to focus its attention. This eliminates the need for complex image or filter pyramids. Secondly, the Faster RCNN introduces the concept of anchor boxes, which are reference boxes with specific scales and aspect ratios. These anchor boxes create a pyramid of choices for each region, allowing the detection of objects at different scales. Lastly, sharing convolutional computations between the RPN and Fast R-CNN reduces computational time. In essence, Faster R-CNN builds upon its predecessors by incorporating the RPN, anchor boxes, and shared computations, resulting in a faster and more effective object detection system (Gad, 2020).

2.7 Evaluation Metrics

The evaluation metric that will be used in this project are confusion matrix, precision, recall, precision-recall curve, intersection over union (IoU), average precision and average mean precision (mAP). The table below shows the definition of the term used in evaluation metric.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

The number of positive samples that were correctly identified is represented as TP, the number of negative samples that were incorrectly identified is represented as FP, and the number of negative samples that were accurately identified is represented as TN. The number of positive samples that were incorrectly identified as negative samples is represented as FN. However, True Negative will not be used in the performance metric because it suggests that the model correctly predict that the data does not contain the target which is not the objective for the project as the object objective is to accurately detect the plastic bottles. The confusion matrix is useful for the evaluation of the performance for the deep learning models. In this project, it is classified as TP when the plastic bottles are correctly identified, FP when the identified items are plastic bottles and FN when the plastic bottles are not correctly identified.

2.7.1 Precision

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

Precision refers to a model's capability to accurately detect relevant items, measured as the ratio of accurate positive predictions. The precision can be calculated by dividing the total number of true positives by the sum of the number of true positives and false positive, and it shows the relationship between correct detections and total detections.

2.7.2 Recall

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

Recall represents a model's ability to locate all pertinent instances, quantified as the ratio of correct positive predictions out of all actual relevant cases. Recall is measured by dividing the total number of true positives by the total number of true positives plus false negatives. It demonstrates the ability to identify true positives in samples without accounting for false positives.

2.7.3 Precision-Recall Curve

The relationship between precision and recall for various confidence threshold values is shown by a precision-recall curve. Both the precision and recall need to be as high as possible, hence, using a precision-recall curve can see which confidence threshold works best for the deep learning model.

2.7.4 Intersection Over Union (IoU)

$$Intersection\ Over\ Union = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Intersection over Union (IoU) stands as a widely used performance metric for accessing the accuracy of object localization and determining localization errors in object detection models. IoU is calculated by dividing the region that is covered with both the predicted bounding box and the ground truth bounding box by the total area covered by both boxes combined. It measures how much overlap there is between two bounding boxes, one of which represents the estimated location and the other the actual location. IoU stands for the percentage of the combined area of two bounding boxes that is shared between them. Both the anticipated and ground truth bounding boxes encompass the area of the union, which is used as the denominator in this calculation.

2.7.5 Average Precision (AP)

$$Average\ Precision = \int_{r=0}^{1} p(r)dr$$

Average Precision (AP) serves as a crucial performance metric aimed at reducing the reliance on a single confidence threshold selection. It is calculated as the area under the Precision-Recall (PR) curve. AP essentially condenses the entire PR curve into a single numerical value. When both precision and recall are high across various confidence threshold values, the AP is also high. Conversely, if either precision or recall is low throughout this range, the AP will be low. The possible range for AP spans from 0 to 1, providing a concise assessment of a model's object detection performance without being tied to a specific confidence threshold.

2.7.6 Mean Average Precision (mAP)

$$mAP = \frac{1}{k}\sum_{i}^{k} AP_i$$

The Mean Average Precision (mAP) is a measurement used to evaluates the performance of object detection models. It involves computing the average precision (AP) for each of these models and then finding the mean of these average precision (AP) values. These calculations take into account a recall range of 0 to 1. In the evaluation of object detection algorithms, the mean average precision (mAP) measure is frequently used. It provides a comprehensive assessment of a model's correctness while taking into account varying degrees of precision and recall. In particular, mAP is preferred for evaluating the effectiveness of object detection models because it takes into account the trade-off between precision and recall and provides a thorough assessment of the model's capabilities.

2.8 Related Works

No	Title	Author	Problem Statement	Technique	Result	Remarks
1.	Object Detection on Bottles Using the YOLO Algorithm	F. S. P. Akbar, S. Y. P. Ginting, G. C. Wu, S. Achmad and R. Sutoyo (2020)	1. Waste such as plastic bottles has become one of the biggest problems for humans. 2. Human labor is low efficient and high operation cost.	1. Dataset: COCO dataset 2. YOLO v2 3. YOLOv3	1. Comparison between YOLOv2 and YOLOv3: The YOLOv2 F1 Score is 0.88, with precision at 1 and recall at 0.79. The YOLO v3 F1-Score is 0.88. Recall score is 0.81, while accuracy score is 0.96. (YOLOv3 has higher recall score than YOLOv2, but lower precision	1. COCO dataset contain 8880 labels for bottles for model training. 2. Using same programming language with the deep learning to code the program is search to make object detection application.

					score compared with YOLOv2.)	
2.	YOLO-based Network Fusion for Riverine Floating Debris Monitoring System	N. A. Zailan, A. S. Mohd Khairuddin, U. Khairuddin and A. Taguchi (2021)	1. Riverine floating debris has long been a significant global challenge. 2. Traditional methods, such manual counting, may require a lot of labor and may not be consistent	1. Dataset: MS-COCO dataset 2. YOLOv4	1. In terms of training duration, mean average precision (mAP), F1 score, average IoU, precision, and recall, detection system performance with transfer learning outperforms detection system performance without transfer learning. 2. In terms of mean average precision	1. Trasnfer learning on MS-COCP Datset for YOLOv4 to improve the detection accuracy and the reduce time taken for the model to be trained. 2. SPP is adopted in the YOLOv4 over CSPDarknet53

			between survey sites. 3. Previous debris detection systems have been restricted to a limited number of object classes and have not attained satisfactory outcomes in terms of both accuracy and		(mAP), F1 score, average IoU, precision, and recall, detection system performance with picture augmentation outperforms that of the system without image augmentation. 3. The suggested approach performs best on average at the IoU threshold of 0.3, where classification accuracy and	

			speed of execution.		precision are the highest for all classes at 74% and 78%, respectively.	
3.	Artificial intelligence (AI) application on plastic bottle monitoring in coastal zone	Do, H.T. and Thi, L.P. (2020)	1. Uncollected plastic bottle waste moves from ocean back to continent by waves causing environmental problems to coastal zone. 2. There is a lack of knowledge in	1. Dataset: Pascal Visual Object Classes (PASCAL VOC) 2. YOLOv3	1. In average, the YOLOv3 could detect 68.72% of plastic bottle in the image for uniform and noise background when only one plastic bottle in a photo. 2. The detection ability of the YOLOv3 had significant decrease to 50% in noise environment when	1. Transparent plastic bottles are harder to be detected compared with coloured plastic bottles. 2. The use of AI in monitoring plastic bottle waste is more effective than human monitoring. 3. The detection ability of the YOLOv3 is better by detecting

			utilizing AI for environmental monitoring, particularly in the context of monitoring plastic bottle waste in coastal zones		two or more plastic bottles in a photo while the detection ability of the YOLOv3 is 63.33% for uniform background. 3. 72.9% of clear plastic bottles in the photos could be detected, but only 50% of unclear plastic bottles can be detected in photos. 4. The detection results demonstrated that the YOLOv3 could successfully identify	from video compared to photo.

					100% of single plastic bottles and 96.05% of multiple bottles in videos, regardless of whether the background was uniform or noisy.	
4.	Deep learning networks for real-time regional domestic waste detection	Mao, W.L., Chen, W.C., Fathurrahman, H.I.K. and Lin, Y.H. (2020)	1. The detection model trained on a dataset with only one object was inadequate for sorting multiple waste objects.	1. Dataset: TRWD, TrashNet 2. YOLOv3	1. TRWD-trained Yolo-v3 achieved mean average precision (mAP) using 0.5 IOU threshold of 92.12% and could detect waste in real-time which is higher	1. The waste type in the TRWD can be further subdivided (Example: Divide the category metal to iron and aluminium) in

					than the performance of TrashNet-trained YOLOv3 which the mean average precision (mAP) using 0.5 IOU threshold is 81.3%.	future for better performance in waste detection.
	YOLO-Green: A Real-Time Classification and Object Detection Model Optimized for Waste Management	W. Lin (2021)	1. Unrecycled solid wastes subsequently pollute the environment directly, posing a threat to both the health of the planet and	1. Dataset: TrashX, TrashNet 2. YOLO-Green that based on YOLOv4 3. ResNet-50 4. DenseNet-121 5. SSD300	1. YOLO-Green stand out from other YOLO models in term of mean average precision (mAP) with 78.04% and achieved 117 MB model size, 12 hours training time	1. YOLO-Green is a modified version of YOLO-v4 with a streamlined architecture. It reduces parameters for bounding box predictions by removing some convolutions in each

			its long-term sustainability 2. Waste management relies on manual labor. 3. Current popular deep learning models (YOLO, DenseNet, ResNet and SDD) are not good fits for trash classification and the	6. YOLOv3 7. YOLOv4	and 2.72 frame per second (FPS).	block from YOLOv4. It maintains the initial four convolutional layers of YOLOv4 in the front part, incorporates two upsampling and two unique downsampling steps in the middle, adds fire modules and convolutional layers for parameter reduction, and merges feature maps at strategic points. The final part

			detection accuracy and speed is low.			includes densely connected fire modules and detection layers. 2. The dataset in TrashX and TrashNet is combined and divided to seven most common types solid trash (batteries, clothes, e-waste, glass, metal, paper and plastic).
6.	Yolo-Based Multi-Model Ensemble for Plastic	L. Liu, B. Zhou, G. Liu, D. Lian and	1. Floating plastics present a significant	1. YOLOv5n 2. YOLOv5s 3. YOLOv5m 4. YOLOv5l	1. The findings indicated that using YOLOv5 with a larger model size	1. Ensemble modeling combines predictions from individual base

	Waste Detection Along Railway Lines	R. Zhang (2022)	hazard to the safe operation of high-speed trains.	5. YOLOv5x	(YOLOv51) yielded superior results, achieving an overall accuracy of 82.6% and a mean average precision (mAP) of 0.822. 2. The ensemble-1 model that use YOLOv5n, YOLO5s and YOLOv5m as the base models achieved an overall accuracy of 83.6% and a mean average precision(mAP) of 0.822.	models to produce a final prediction and it can improve the performance of object detection effectively. 2. The YOLO-based ensemble model performed better in the daytime condition compared to nighttime condition.

					3. The ensemble-2 model that use YOLOv5n, YOLO5s, YOLOv5m and YOLOv51as the base models achieved an overall accuracy of 85.4% and a mean average precision(mAP) of 0.834.	
7.	Research on Improved Yolo on Garbage Classification Task	Z. Pan (2022)	1. The urgent of waste classification treatment as more waste is produced	1. YOLO v3	1. The average performance of the optimized version YOLOv3 (YOLOv3++) is 0.12% better than	1. The improved version of the YOLOv3 (YOLOv3 ++) include optimizing the backbone by

			with the development of society.		Yolov3 and the detection time for YOLOv3++ is 0.6 seconds faster than YOLOv3.	adjusting the input image size from 608 to 618 to enhance feature scale, replacing the RELU activation function with Leaky RELU to prevent data explosion, optimizing the Darknet 52-layer model to 43 layers to reduce computation, and using the K-means clustering algorithm to enhance model accuracy. Additionally, the

						output feature sizes are adjusted from 38x38 and 76x76 to 19x19, further enhancing model suitability. Transfer learning with ImageNet parameters is also applied to boost detection accuracy.
8.	Design of Plastic Bottle Image Recognition System Based on	J. Xiao, Y. Tang, Y. Zhao and Y. Yan (2020)	1. Garbage classification in China still mainly relies on manual classfication which is low	1. YOLOv3	1. The YOLOv3 model achieved 91.3% for the accuracy and 26 frame per second (fps) for dynamic detection speed.	1. The low light environment will affect the recognition accuracy due to insufficient dataset.

	Improved YOLOv3		efficiency and unsafe.			
9.	Development and Testing of Garbage Detection for Autonomous Robots in Outdoor Environments	Y. Arai, R. Miyagusuku and K. Ozaki (2021)	1. Labour shortage for garbage collection in Japan due to the declining of the birthrate and aging population.	1. YOLOv2	1. YOLOv2 achieved an average precision with 92.9% and average recall with 89.9% for all classes (can, PET and lunch box).	1. Plastic bottles are challenging to detect due to their transparent material. 2. Objects with similar features, such as cans and plastic bottles, may encounter misdetection issues. This problem can be addressed by including more images of cans in

						the dataset, as they are often misclassified as plastic bottles.
10.	Robotic Detection of Marine Litter Using Deep Visual Detection Models	Fulton, M., Hong, J., Islam, M.J. and Sattar, J. (2019)	1. Shallow water is affected by varying light conditions, and the presence of turbid water can make detecting	1. YOLOv2 2. Tiny-YOLO 3. Faster RCNN with Inception v2 4. Single Shot MultiBox Detector (SSD) with MobileNet v2	1. Faster R-CNN has the highest mAP of 81.0 compared to YOLOv2, Tiny-YOLO and SSD but the it has higher inference time compared to YOLOv2 and Tiny-YOLO. 2. YOLOv2 have a good balance	1. YOLOv2 performance can be further improved by other methods. 2. Faster R-CNN and SSD are less suitable than YOLO for real-time plastic bottle detection in terms of the balance between

			objects difficult or even impossible. 2. Marine debris rarely remains in pristine condition, degrading over time.		between accuracy and speed. 3. SSD has the best inference times compared to other models. 4. Tiny-YOLO has the best performance on TX2 which is the most realistic hardware for a modern autonomous underwater vehicles (AUV).	accuracy and inference time.
11.	Garbage Classification System with	Yang, G., Jin, J., Lei, Q., Wang, Y., Zhou, J., Sun,	1. The challenge of efficient garbage classification	1. Dataset: TACO 2. YOLOv5	1. The mean Average Precision (mAP)in all classes using YOLOv5l is 94.5%	1. YOLOv5s is more suitable for larger target detection but not for small targets

	YOLOV5 Based on Image Recognition	Z., Li, X. and Wang, W. (2021)	in China is highlighted by issues like citizen participation, accurate sorting, and enforcement difficulties, given the country's large population. Law enforcement officers face challenges in efficiently		after using the designed batch size 24 and 50 epochs for training.	like garbage due to its focus on speed.

			and accurately separating mixed garbage during the collection process of waste for a 1.4 billion population country.			
12.	A multi-label waste detection model based on transfer learning	Zhang, Q., Yang, Q., Zhang, X., Wei, W., Bao, Q., Su, J. and Liu, X.	1. The majority of citizens are unfamiliar with waste classification	1. Dataset: MULTI-TRASH dataset 2. YOLO-waste 3. YOLOv4	1. The experimental results indicate that the YOLO-WASTE model achieves an mAP value of 93.12% and can	1. The MULTI-TRASHdataset used in this study is relatively small, making the trained waste detection

			standards and specific rules. 2. Traditional waste sorting methods and indirect waste sorting have worse performance in waste sorting than deep learning. 3. Many waste image datasets primarily consist of individual		detect an image in an average time of 0.424 seconds. 2. The YOLO-WASTE model achieved 94.50% for the precision score, 92.22% for recall score and 93.33% for F1-score in total target item.	model susceptible to overfitting and limiting its ability to benefit from deep learning technique，thus transfer learning is used to solve the problem of insufficient traing data. The YOLOv4 multi-label dettection model is pre-trained by using PASCAL VOC dataset before using the multi-label waste image dataset.

			waste images, which do not adequately represent real-life scenarios where various types and quantities of waste are mixed together.			2. A single image in a dataset should consists of multiple waste to stimulate the real world situation.
13.	Detection and Location of Domestic Waste for Planning Its	P. Tornero, S. Puente and P. Gil (2022)	1. The EU governments want to keep improve the quantity of	1. YOLOv3-tiny 2. YOLOv3-SPP 3. YOLOv5s	1. YOLOv5l was chosen because it outperforms other models, achieving a mean average	1. False positive can be reduced by increasing the confidence threshold.

	Collection Using an Autonomous Robot			waste that was being recycled to decrease municipal waste landfilled.	4. YOLOv5l	precision (mAP) of 0.9951 at an IoU threshold of 0.5 and an mAP@.95 of 0.8424 during the best training. Additionally, it can detect instances in an image in an average time of 36 ms. 2. The location error is greater, and its variation is more pronounced when the measures exceed the range of 0.3 to 3 meters since the	

					maximum measuring range for the RGBD camera is approximately 3 meters.	
14.	An autonomous robotic system for collecting garbage over small water bodies	S. N. Hasany, S. S. Zaidi, S. A. Sohail and M. Farhan (2021)	1. Inproper disposable plastic bottles contribute significantly to plastic waste and often end up polluting water bodies. 2. Most prototypes for garbage	1. Tiny YOLO	1. The Tiny YOLO model can achieve mAP of 86.9% for the plastic bottle detection and the robot is able to reach and collect the plastic bottle waste with minor miscollection.	Time taken for models to predict is depends on the size of the model. Tiny Yolo was choosen because of its three times smaller to original YOLO detector.

			collection are not autonomous and require an operator to control them.			
15.	Detection and Classification of Floating Plastic Litter Using a Vessel-Mounted Video Camera and Deep Learning	Armitage, S., Awty-Carroll, K., Clewley, D. and Martinez-Vicente, V. (2022)	1. The use of boat-mounted cameras in detecting the plastic debris will generates vast amount of data which can be time-consuming to analyze.	1. Data processing: Video and Image Analytics for Multiple Environements (VIAME) software 2. YOLOv5m 3. YOLOv5s	1. YOLOv5s that used image size of 1280 pixels was selected to be used due to the high accuracy and low computational input even though YOLOv5m has slighly higher accuracy. This is due to the	1. Accuracy of the YOLO models can be improved by increasing the image size. 2. Trade off between computational resources and accuracy.

					2. YOLOv5s can achieved the accuracy of 95.23% after training when only detect the presence or absence of plastics. 3. However, the model acuracy will has significant decrease to 65.3% when differentiating between three predefined categories(plastic bag, plastic bottle or plastic buoy)	

16.	YOLO-based robotic grasping	M. Kim and S. Kim (2021)	1. The Generative Residual Convolution Neural Network (GR-ConvNet), does not have recognition function. 2. GR-ConvNet has problem on grasping things outside of the field of view.	1. Dataset: Open Image Dataset (OID) 2. YOLOv5x 3. YOLOv5s	1. YOLOv5s has better performance compare to YOLOv5x for the waste detection, in term of precision, recall and mean average precision (mAP)	YOLOv5 is used to solve the recognition problem without having signifcant increased on the inference time.

17.	Object Detection for Construction Waste Based on an Improved YOLOv5 Model	Zhou, Q., Liu, H., Qiu, Y. and Zheng, W. (2022)	1. The conventional approach to sorting construction waste involves a combination of mechanical processes like mixing, crushing, and screening, alongside manual labor for preselection, rejection, and	1. Dataset: PASCAL VOC 2. YOLOv5 3. Improved YOLOv5 model	1. The improved YOLOv5 has higher detection accuracy and the mean average precision (mAP) can reach up to 0.9480.	1. TrashNet and Taco, open-source datasets do not suitable for robotic sorting system because the objects detected and transferred on a conveyor belt tend to be irregular, dirty, and stacked on top of each other. 2. An improved YOLOv5 model was proposed, which involved using the CBAM attention mechanism and SimSPPF module,

			diversion. However, this method faces challenges such as low recycling purity, inefficient manual work, and significant health risks in dusty and noisy environments.			adding a layer for detecting small construction waste objects, addressing inter-occlusion, and enhancing feature fusion at the fourth scale. 3. CBAM-CSPDarknet53 and multi-scale detection are used to detect small object in the image. 4. The SimSPPF module prevents local feature loss in construction waste

						dataset images, efficiently decreases residual parameter information, and preserves essential texture features. It also speeds up forward propagation compared to the SSP module.
18.	Improved YOLO Based Detection Algorithm for Floating Debris in Waterway	Lin, F., Hou, T., Jin, Q. and You, A. (2021)	1. Traditional image processing methods struggle to fulfill the demands of real-time	1. SSD 2. YOLOv2 3. YOLOv3 4. YOLOv4 5. YOLOv5s 6. YOLOv5m 7. FMA-YOLOv5s	1. FMA-YOLOv5s has the highest mAP compared to other models which is 77.83% in original dataset and 79.41 in expanded dataset.	1. Expanding the number of images for training set can increase the mAP of the models.

			monitoring for floating debris in waterways due to environmental complexities.		2. The mAP value of FMA-YOLOv5s exceeds YOLOv5s by 2.18% with only 1 FPS lower than YOLOv5s.	2. The FMA-YOLOv5s model which based on YOLOv5s added a feature map attention later at the end of the backbone to improve the ability of feature extraction.
19.	A Comparison of RGB and RGNIR Color Spaces for Plastic	Tamin, O., Moung, E.G., Dargham, J.A., Yahya, F., Omatu, S.	1. RCNN algorithm takes a longer time and pose optimization challenges	1. Dataset: Red-Green-Blue (RGB) channel images, Red-Green-Near-	1. The YOLOv5m achieved a mean weighted metric score(WMS) of 70.79% and 71.72% for RGB images and	1. K-Fold Cross-validation is employed to provide a more representative assessment of

	Waste Detection Using The YOLOv5 Architecture	and Angeline, L. (2022)	since individual training is required for each component within the image. 2. The post-processing algorithm has a tendency to misclassify background patches as objects, primarily due to its	Infarred (RGNIR) images 2. YOLOv5m	RGNIR respectively during validation and achieved 70.07% and 71.78% for RGB images and RGNIR respectively for testing dataset.	the entire dataset. The higher the number of cross-validation folds, the less bias there is towards overestimating the true expected error. However, there is a trade-off involving accuracy and computational power, as more computational resources are needed for

			restricted context awareness.			higher fold counts. 2. RGNIR images dataset give a better representation for the object detection models.
20.	YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins	Wahyutama, A.B. and Hwang, M. (2022)	1. People do not recognize the significance of waste separation and do not dispose of their waste	1. YOLOv4	1. YOLOv4-Tiny can achieved an accuracy of 97% to 99% implemented in Raspberry Pi in actual scenario. 2. Full-size YOLOv4 has higher mAP of	1. YOLOv4-tiny was still chosen due to the Raspberry Pi's performance constraints and the need for miniaturization.

			according to its category.		91% than YOLOv4-Tiny.	
21.	Research on solid waste plastic bottle cognitive based on YOLOv5s and deep stochastic configuration network	Chen, K., An, J., Fang, Y. and Bu, T. (2022)	1. Environment complexities caused the identification of plastic bottles using image recognition and target detection to be difficult. 2. Although the improved DenseNet121 algorithm has high	1. YOLOv5s 2. DeepSCN	1. YOLOv5s+DeepSCN achieved the highest average recognication rate of 98.60% compared to YOLOv5s+softmax, YOLOv3 and SSD. 2. The training time for YOLOv5s+DeepSCN is also the lowest compared to other algorithms which is 0.786h.	1. The introduction of SiLU activation function reduces the computation,saves storage space and effective in solving overfitting problem. 2. The modification to the network structure of YOLOv5s, which

			accuracy, it takes a long time to detect, struggles to recognize similar objects in bright light, and has trouble with stacked garbage, all of which can affect classification results.			incorporates a stochastic configuration network classifier into the conventional convolutional neural network, allows for obtaining more detailed characterization of solid waste images at a lower feature level. This makes it easier to build a classification

						system for domestic solid waste. 3. The use of DeepSCN significantly reduces the randomness in results caused by the initial random selection of network weights and biases, resulting in a more stable and generalized performance.

22.	Visual Detection of Waste using YOLOv8	R. Bawankule, V. Gaikwad, I. Kulkarni, S. Kulkarni, A. Jadhav and N. Ranjan (2023)	1. The amount of focus on environmental conservation is rising. 2. Pollution of water and air, transmission of diseases, lack of facilities in waste management system	1. YOLOv8 2. YOLOv7 3. YOLOv5 4. YOLOv4 5. Faster R-CNN 6. SSD	1. YOLOv8 achieved the highest mean Average Precision (mAP) among all the deep learning models which is 97.7%.	1. YOLOv8 shows high accuracy in classifying waste product compared with other existing object detection algorithm existed. YOLOv8 is good option for model training in this project to choose for detecting plastic waste products.

2.9 Summary

The insights gained from these research papers offer valuable guidance and inspiration for developing applications for the PET-plastic bottle detection using deep learning. They highlight the pivotal role of advanced technology in addressing the increasingly pressing issues surrounding plastic waste management, plastic pollution control, and efficient plastic waste sorting. These studies underscore the limitations of manual waste classification processes, including low efficiency, high operational costs, labour shortages, and the lack of comprehensive knowledge in handling the mounting volume of plastic waste in modern society. This awareness is a compelling motivation for developing automated solutions for plastic waste sorting based on deep learning models.

Thus, these research papers introduced deep learning, particularly the YOLO (You Only Look Once) family of models as YOLO models have emerged as a promising solution for tackling waste detection and classification challenges. Various versions of YOLO, such as YOLOv2, YOLOv3, YOLOv4, and YOLOv5, have been explored in these papers, each with its own advantages and trade-offs. These deep learning models have demonstrated impressive capabilities in terms of accuracy, speed, and real-time performance, which makes them a suitable solution for plastic bottle detection.

In addition, there have been discussions about incorporating faster CNNs (Convolutional Neural Networks) and Single Shot MultiBox Detectors (SSD) into the research landscape. Faster CNNs have shown better accuracy, although they come with longer inference times, while SSD has demonstrated better inference time at the cost of slightly lower accuracy. These alternative approaches offer interesting avenues for further exploration in the context of plastic bottle detection. Hence, this makes YOLO a favourable choice for real-time applications as it strikes a balance between inference time and accuracy.

Furthermore, the techniques introduced in these papers, such as transfer learning, image augmentation, and ensemble modelling, offer practical approaches to enhance the performance of deep learning models. These strategies can be incorporated into the development process of plastic bottle detection applications, resulting in more reliable and robust systems that accurately identify plastic bottles even in varying environmental conditions.

However, the challenges outlined in these studies should not be underestimated. Environmental complexities, including fluctuating light levels and the diverse states of plastic bottles waste, pose significant obstacles to visual detection methods. To tackle these challenges effectively, this application development for plastic bottle detection need to consider how the solution can adapt to changing environmental conditions and recognize plastic bottles in various states of degradation. Additionally, selecting an appropriate dataset plays a pivotal role in the training of deep learning models. The studies recommend using larger, more diverse datasets that closely resemble real-world waste situations. An extensive and representative dataset will be valuable in training deep learning models to accurately detect plastic bottles.

Finally, selecting the appropriate pre-trained deep learning models should be based on the specific goals of the plastic bottle detection application. Depending on the context, developers might prioritize either accuracy, speed, or resource efficiency. Choosing the suitable pre-trained deep learning model to the application's needs is crucial for achieving optimal performance in detecting the plastic bottles on the moving conveyer belt in the recycling centres. Some deep learning models emphasize speed, while others focus on accuracy, and the choice should align with the requirements of the project. For instance, YOLOv5s may be more suitable for detecting larger targets, while YOLOv4-tiny might be the preferred option for resource-constrained environments such as Raspberry Pi.

In summary, the insights gained from these research papers provides a foundation for the development of applications on plastic bottle detection using deep learning. By leveraging the experiences and techniques shared in these research paper, it is possible to create an innovative solution that contributes to the more efficient sorting of plastic waste in recycling centres, thus reducing plastic pollution and create a cleaner, more sustainable environment to the society.

CHAPTER 3

PROJECT WORKPLAN AND TOOLS

3.1 Introduction

This chapter outlines the approach and instruments utilized in the project. It begins by examining the Kanban system, known for its effective work organization. The project explores Kanban flexibility, applicable not just to project management but also to personal growth and deep learning model training. The project progresses through specific steps, starting with concept identification and concluding with deployment. Each step is carefully planned to ensure the plastic bottle finder functions effectively. The project also introduces essential tools like Roboflow, Google Colab, and Gradio, which assist with tasks such as data preparation, deep learning model training, and user interface development. Finally, a structured overview of project tasks and activities is presented through a Work Breakdown Structure (WBS).

3.2 Project Methodology

Kanban is a popular agile methodology that used in software development. The concept of Kanban methodology had been introduced by Corey Lada in 2009 and David Anderson in 2010. David Anderson's 2010 book elaborated on how Kanban might serve as the foundation of an effective, flow-based software development strategy (Anderson, 2010) while Corey Ladas' 2009 book established how the Kanban methodology in tracking and helping work in progress within software development (Ladas, 2009). Kanban methodology uses Kanban Board to visualise the project progress by using columns to represent different stages of a project. Kanban cards are used to represent work items or task and move from column to column after the tasks completed. Thus, Kanban is good for providing an overview of the project progress. In addition, Kanban is a flexible framework which allows tasks added to the Kanban board at any time. Furthermore, Kanban methodology promotes continuous feedback which allows improvement for the deep learning model or the application development. This makes Kanban methodology suitable for this project.

Kanban is an agile process methodology without timeboxes. This makes Kanban suitable for deep learning-based project as it is hard to define when the deep learning process is complete. The deep learning model training process often ended when the development team feels that the accuracy of the model had reached the optimize solution, thus making it hard for the team to estimate the duration needed for deep learning model training. Furthermore, more techniques to optimize the performance of the deep learning models may be introduced in the middle of the training process. With the flexibility provided by Kanban methodology, the development team can add any new task to the Kanban board for the project. In addition, compared Kanban with Scrum, it is hard for the development team to scope for the deep learning-based project in advance because the performance result of the deep learning model is unknown until the deep learning model finished training, making Kanban a better methodology to be used in deep learning-based project.

In brief, Kanban is chosen in this project because it is an agile process methodology that emphasizes improvement. Besides, flexibility of Kanban methodology allows any task to add on to the project whenever which phases of the project is. The versatile of the Kanban methodology compared to Scrum making it suitable for this project because work need to be completed in batches if Scrum methodology is used. Additionally, due to different column in Kanban board which breaks down the task into “To Do”, “On progress” and “Completion” helps to keep track of the development process to make sure every task is completed and did not miss out any of the important task. Furthermore, Kanban methodology also provides feedback which helps to improve the development of the application. As the performance of the model can be viewed after the model training process ended, the development team can straight finetuning to the model if the result is not satisfied instead of completing every task for the project, then back to the optimizing process.

3.2.1 Phase 1: Concept Identification

With growing concerns about plastic bottles waste and its harmful impact on the environment, this project aims to use State-of-Art deep learning models to reduce the plastic bottles waste pollution to the environment. In the initial phase, the research will be conducted to gain a comprehensive understanding of plastic bottle detection using various types of deep learning models. This research will serve as the foundation for creating a solution that can identify PET-plastic bottles accurately. By addressing this problem, the project demonstrates the potential of deep learning model in handling plastic bottle sorting issues and provides efficient solutions for sorting in recycling factories.

3.2.2 Phase 2: Project Inception

The development of this plastic bottle detection application begins with a well planning to ensure the project works smoothly. The project begins by gathering the functional and non-functional requirements needed for the project. A use-case diagram is created to show an overview functionality of the application and the entity interact with the application. Additionally, detailed descriptions of each use case are created to illustrate how the application will function.

Furthermore, a comprehensive work breakdown structure is developed to ensure the project proceeds smoothly by breaking it down into manageable tasks and activities. The project then proceeds to the selection and training of deep learning models using YOLOv8, Faster RCNN, and SSD to compare their performance in accurately detecting plastic bottles. Additionally, the project team identifies and incorporates relevant tools, including data annotation tools (Roboflow) and model training tools (Kaggle and Google Colab), to support the project's objectives.

To provide a clear overview of the project's workflow, a flow chart is included, outlining the sequential steps and processes involved in developing this plastic bottle detection application. This systematic approach ensures that the project is well-organized, efficient, and capable of achieving its goal of effectively detecting plastic bottles while maintaining user-friendliness.

3.2.3 Phase 3: Iteration

In this project, Kanban methodology will be followed and the first step for the iteration is the dataset preparation. Plastic bottle images will be collected, and once obtained, Roboflow will be utilized as a data annotation tool to label images and export the data in various formats suitable for training deep learning models. The choice of data format exported with depends on the specific deep learning model selected for training.

In the subsequent phase, the focus will shift to the training of the deep learning models which is YOLOv8, Faster R-CNN and SSD. Their performance will be evaluated using mean average precision metrics to validate their ability to detect plastic bottles effectively. To enhance the performance of the models, the optimization for the training of the deep learning models will be repeated to improve detection accuracy of the plastic bottles. If the models' performance falls short of expectations, the training process will be iterated until an acceptable level of accuracy is achieved.

Ultimately, the project will select the deep learning model with the best performance from the trained deep learning models. This chosen model will be prepared for integration into the application, ensuring its capability to reliably detect plastic bottles. These steps will be repeated if the performance results are not satisfactory.

3.2.4 Phase 4: Development and Testing

The user interface will be developed for the selected deep learning model to ease the end-user who does not have the knowledge in the deep learning. This user-interface ensures that end-users won't need to deal with the complexities of the deep learning model.

Following the development of the user interface for the selected deep learning model, a series of tests such as unit testing, integration testing, usability testing and user acceptance testing will be conducted to the application. These tests will verify that the application functions as expected and fulfils all the specified requirements. Any bugs or issues identified during testing will be addressed and resolved in this phase.

3.2.5 Phase 5: Deployment

After ensuring the plastic bottle detection application works effectively without any problems, it is prepared for deployment. Once all the necessary configurations for the application are ready, it will be passed to the engineering faculty in Universiti Tunku Abdul Rahman for future work. The model is expected to integrate with the robotic arm to detect plastic bottles and pick the plastic bottles on the moving conveyer belt in the recycling centres. The application is expected to perform real-time plastic bottle detection and has the potential to replace manual sorting in the recycling centres, thus increasing the efficiency of plastic bottle sorting processes.

3.3 Project Tools

3.3.1 Roboflow

Roboflow is selected for this project because it is a computer vision developer framework that provides a wide variety of datasets ready for deep learning model training and allows users to upload custom data to create custom dataset. Furthermore, it also acts as a data annotation tool, allowing users to annotate data and offering image processing techniques to process the custom data that users upload. Moreover, it allows the users to do data augmentation to the training datasets. The entire dataset preparation process can be done using Roboflow and exported in required data format based on the deep learning models. Roboflow also provides various method to export the dataset which allow user to straight import the data to the cloud platform without reuploading to the cloud platform by themselves.

3.3.2 Google Colab

Google Colab provides free computational resources with limited quota that help in this project, as training for the deep learning model requires a lot of computational resources that are expensive. Also, the powerful computational resources provided by Google Colab, such as GPUs and TPUs, can speed up the training process of the deep learning model. Free GPU provided by Google Colab shorten the training process compared with training with CPU.

3.3.3 Kaggle

Kaggle is an online community platform for data science, and it provides an online notebook that gives users 30 hours of computational resources per week to train deep learning models. The free GPU resources provided by Kaggle help to speed up the deep learning model training process. Additionally, Kaggle also offers guidance to help beginner users start the learning process in deep learning.

3.3.4 Gradio

Gradio is an open-source Python library that can create easy-to-use, customizable UI components for the deep learning model, APIs, or any other functions with minimal lines of code. Gradio simplifies the process of interacting with the deep learning model within a web browser by allowing users to effortlessly drag and drop elements such as images, text, or even voice recordings. This facilitates real-time, interactive visualization of the results, and users have the option to seamlessly integrate this graphical user interface directly into a Jupyter notebook or share it with others via a link. Gradio will be used in this project to develop the user interface for the pre-trained deep learning model after selecting the deep learning model with the best performance.

3.3.5 Visual Studio Code (VS Code)

Visual Studio Code (VS Code) is a code editor developed by Microsoft. It will be used to develop the user interface for the application and use to integrate the deep learning model with the user interface after the user interface successfully developed. Visual Studio Code is chosen because it has rich extension library make it a valuable tool for streamlining the development and deployment of deep learning models with user-friendly interfaces.

3.4 Work Breakdown Structure (WBS)

1. Phase 1: Concept Identification
   - 1.1 Thorough Research on Plastic Bottle Detection
     - 1.1.1 Plastic Bottles Waste Pollution and Sorting Concerns Analysis
     - 1.1.2 Deep Learning Models Exploration
     - 1.1.3 Plastic Bottle Detection Challenges Study
   - 1.2 Project Scope Gathering
     - 1.2.1 User Scope and System Scope for Plastic Bottle Detection
     - 1.2.2 Limitation for Plastic Bottle Detection
2. Phase 2: Project Inception
   - 2.1 Project Start and Requirement Collection
     - 2.1.1 Functional Requirement Gathering
     - 2.1.2 Non-Functional Requirement Gathering
   - 2.2 High-Level Application Design
     - 2.2.1 Use-Case Diagram Creation
     - 2.2.2 Use-Case Description Writing
   - 2.3 Work Breakdown Structure Development
     - 2.3.1 Task and Activity Breakdown
   - 2.4 Deep Learning Model Research and Selection
     - 2.4.1 YOLO Model Research
     - 2.4.2 Faster RCNN Model Research
     - 2.4.3 SSD Model Research
   - 2.5 Tool Identification and Setup
     - 2.5.1 Data Annotation Tool Setup
     - 2.5.2 Deep Learning Model Training Tool Setup
   - 2.6 Workflow Chart Creation
     - 2.6.1 Project Workflow Diagram
   - 2.7 Deep Learning Model Prototype Development
     - 2.7.1 Dataset Preparation
     - 2.7.2 Deep Learning Model Prototype Training
     - 2.7.3 Evaluation Metrics Analysis
3. Phase 3: Iteration

3.1 Kanban Methodology Implementation

3.1.1 Kanban Board Setup

3.2 Dataset Preparation

3.2.1 PET-Plastic Bottle Image Collection

3.2.2 PET-Plastic Bottle Image Annotation

3.3 Deep Learning Model Training

3.3.1 Model Training for YOLOv8

3.3.2 Model Training for Faster RCNN R50 FPN

3.3.3 Model Training for SSD Mobilenet V2

3.4 Model Performance Evaluation and Fine-Tuning

3.4.1 Evaluation of Deep Learning Models' performance by using mean average precision metric analysis

3.4.2 Optimization of Deep Learning models

3.4.3 Iterative training of Deep Learning models

3.5 Best Model Selection

3.5.1 Deep Learning Model Performance Assessment

4. Phase 4: Development and Testing

4.1 User Interface Development

4.1.1 UI Design for Deep Learning Model

4.2 Testing Phase

4.2.1 Unit Testing for Login and Start Webcam Modules

4.2.2 Integration Testing for Plastic Bottles Detection

4.2.3 Usability Testing using System Usability Score

4.2.4 User Acceptance Testing

5. Phase 5: Deployment

5.1 Application Performance Verification

5.1.1 Plastic Bottle Detection using webcam

3.5 Gantt Chart

A Gantt chart is created to illustrate the schedule for this project to show the start date, end date, duration and the progress of the completion for each task. The Gantt Chart is attached as "Appendix A: Gantt Chart" for this project.

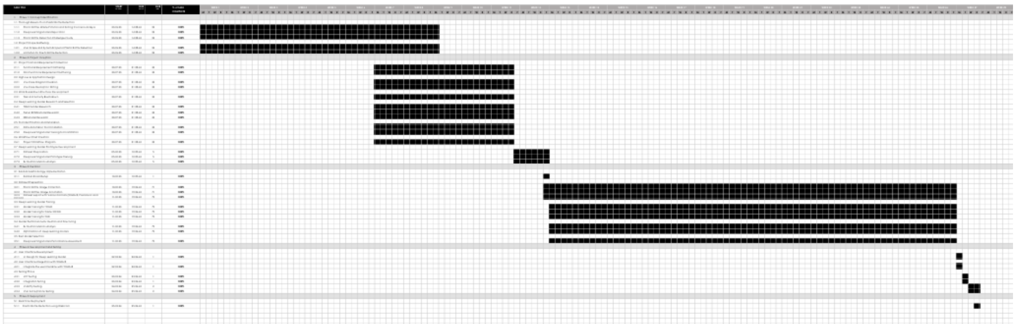


Figure 3.1: Gantt chart

CHAPTER 4

PROJECT INITIAL SPECIFICATION

4.1 Introduction

This chapter explores the specifications and requirements of the plastic bottle detection application. It covers both functional and non-functional requirements for the application. Functional requirements describe how users interact with the application while the non-functional requirements focus on critical aspects beyond functionality, including performance, security, usability, and reliability. These requirements form the basis for a robust and user-friendly application that helps in plastic bottle sorting. Additionally, this chapter provides insights into requirement modelling, featuring use case diagrams and descriptions, along with a detailed system flow.

4.2 Functional and Non-functional Requirements Specification

4.2.1 Functional Requirements Specification

These functional requirements outline the essential features and capabilities of the plastic bottle detection application. They define how users can interact with the application, highlighting the importance of user authentication for security, the real time analysis capability to detect plastic bottles from the webcam, and the user interfaces to interact with the pre-trained deep learning model, initiating detection, and viewing prediction results. These requirements serve as a foundational framework for the development of an efficient and user-friendly application that contributes to reduce plastic bottles pollution. The functional requirements are listed below:

1) The application should authenticate user before logging in to the application to ensure the security of the application.
2) The application shall be able to show prediction to the user and identify the presence of plastic bottles through webcam input.
3) The application shall provide a user interface for user to start and stop webcam on plastic bottle detection.
4) The application shall provide a user interface for user to view detection results.

4.2.2 Non-functional Requirements Specification

Non-functional requirements are a critical aspect of application development, defining the qualities and characteristics that go beyond its core functionality. These requirements encompass factors such as performance, security, usability, and reliability, which collectively shape the user experience and the application's overall success. By setting clear non-functional requirements, developers can ensure that the application not only performs efficiently but also adheres to essential standards and offers a seamless, secure, and dependable experience for the users.

1) Performance:
    - The application shall achieve 1 frame per second for the analysing of the images.
2) Security:
    - The application shall authorize the user before entering the application.
3) Usability:
    - The user interface shall pass the user acceptance test to maximize the user experience by having clear input and output, and responsiveness on various screen sizes.
4) Reliability:
    - The application should be available for all the time during operation hours of the recycling factories.
5) Maintenance:
    - The application should be well-documented to facilitate future updates, enhancements, and bug fixes.

4.3 Requirement Modelling

4.3.1 Use Case Diagram

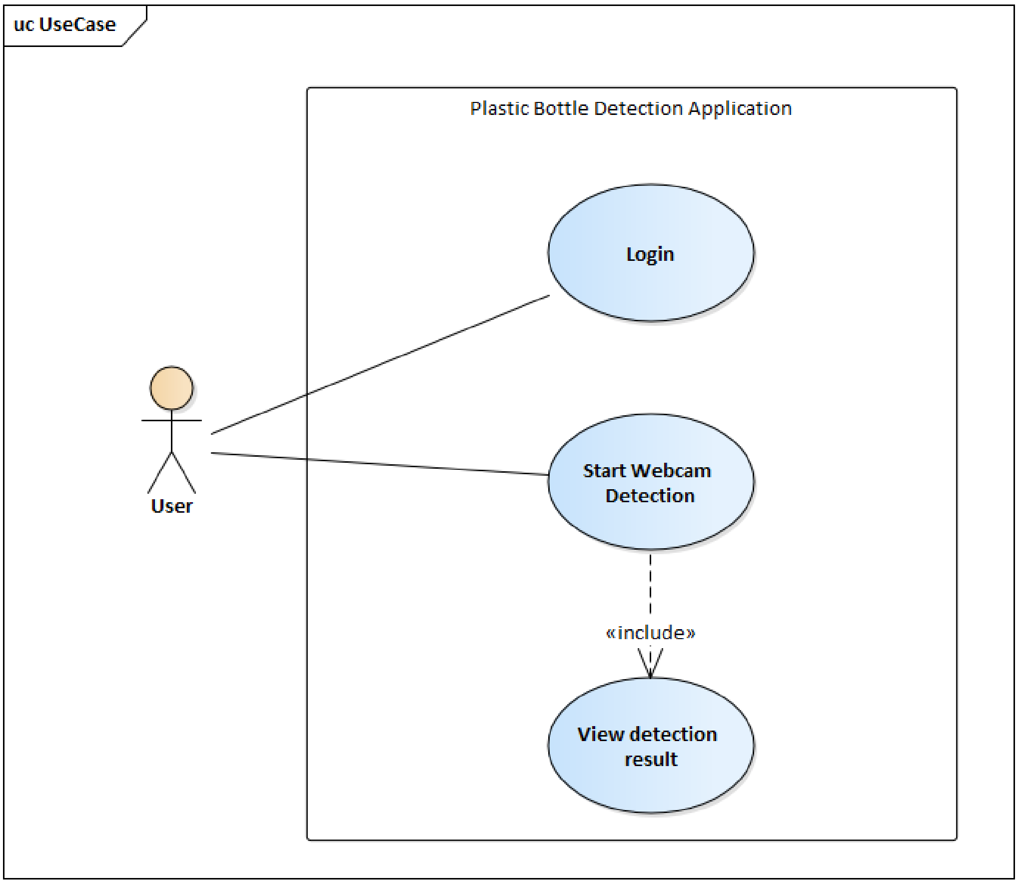


Figure 4.1: Use case diagram

4.3.2 Use Case Description

4.3.2.1 Login

<table>
<tr><td colspan="2">Use Case Name: Login</td><td colspan="2">ID: UC01</td><td>Importance Level: Low</td></tr>
<tr><td colspan="3">Primary Actor: End-user</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="5">Stakeholders and Interests:
End-user-want to login to the application</td></tr>
<tr><td colspan="5">Brief Description:
The use case describes how the end-user login to the application.</td></tr>
<tr><td colspan="5">Trigger: End-user wants to login to the application</td></tr>
<tr><td colspan="5">Relationships:
Association : End-user
Include : N/A
Extend : N/A
Generalization: N/A</td></tr>
<tr><td colspan="5">Normal Flow of Events:
1. The application displays a login page.
2. The end-user fills in the username and password in the respective text field.
3. The application will verify the username and password.
4. The end-user successfully login to the application.</td></tr>
<tr><td colspan="5">Sub-flows:
3.1 If the username and password fill in is correct, the application will allow the end user to login.
3.2 If the username and password fill in is incorrect, the application will prompt the username or password incorrect and let the end-user to key in the information again.</td></tr>
<tr><td colspan="5">Alternate/Exceptional Flows:</td></tr>
</table>

4.3.2.2 Start Webcam Detection

<table>
<tr><td>Use Case Name: Start Webcam Detection</td><td>ID: UC02</td><td>Importance Level: High</td></tr>
<tr><td>Primary Actor: End-user</td><td colspan="2">Use Case Type: Detail, Essential</td></tr>
<tr><td colspan="3">Stakeholders and Interests:
End-user-want to upload image to the application</td></tr>
<tr><td colspan="3">Brief Description:
The use case describes how the end-user upload image to the application.</td></tr>
<tr><td colspan="3">Trigger: End-user wants to upload image to the application.</td></tr>
<tr><td colspan="3">Relationships:
Association : End-user
Include : View detection result
Extend : N/A
Generalization: N/A</td></tr>
<tr><td colspan="3">Normal Flow of Events:
1. The end-user chooses to start the webcam in the application.
2. The application shows the prediction output to the users.
3. The end-user stop the webcam in the application.</td></tr>
<tr><td colspan="3">Sub-flows:</td></tr>
</table>

Alternate/Exceptional Flows: 1a. End-user must successfully log in to the application using valid username and password.

4.4 Proposed System Flow

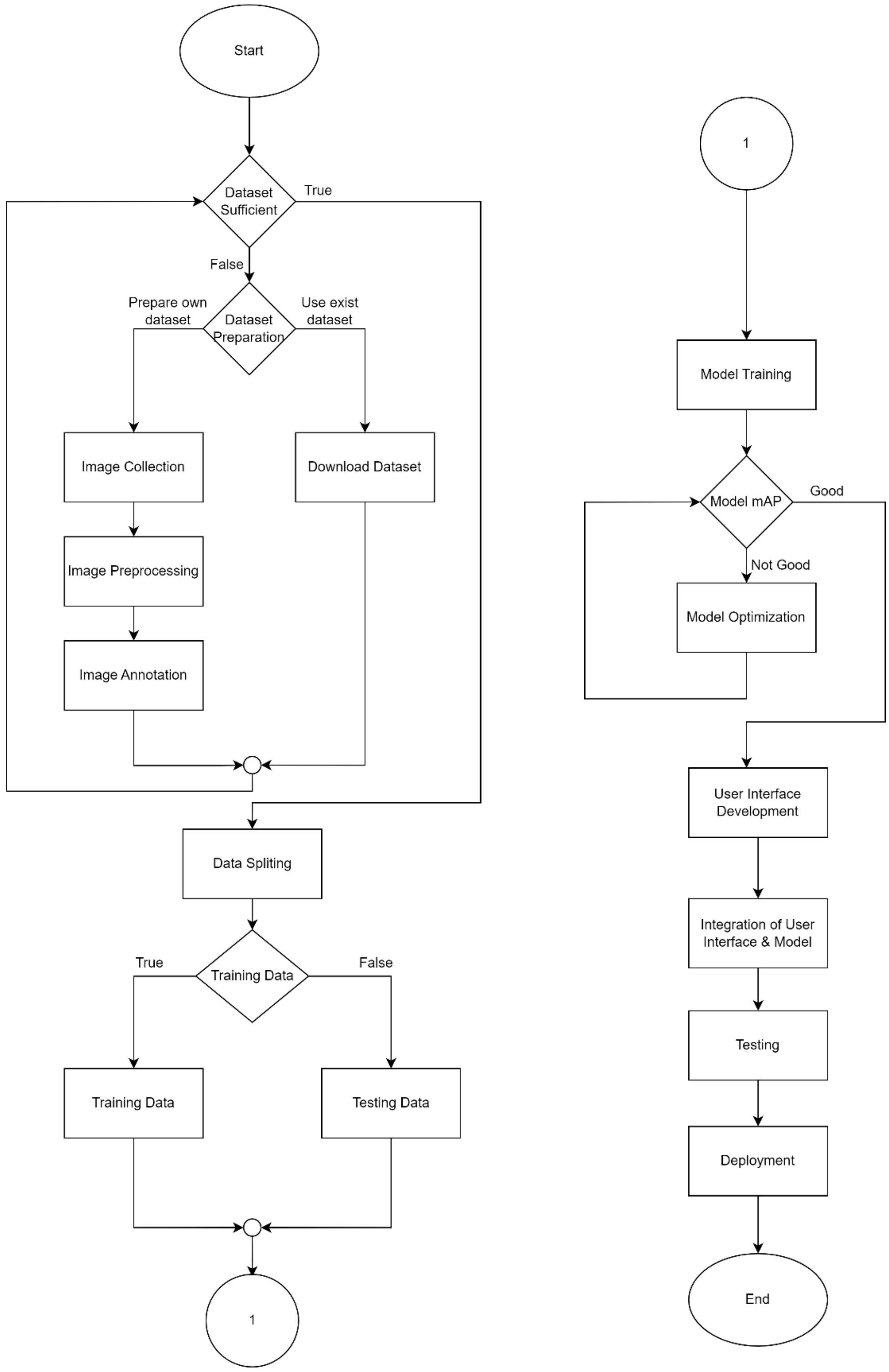


Figure 4.2: Flow chart for project flow

CHAPTER 5

SYSTEM DESIGN

5.1 Introduction

The chapter includes dataset collection, deep learning model training, comparison of deep learning models, and application development.

In this project, three pre-trained deep learning models are used to compare performance in detecting plastic bottles. The selected pre-trained deep learning models are Faster R-CNN, SSD MobileNet V2, and YOLOv8. The items being detected for this project are PET-plastic bottles ranging in size from 500ml to 1000ml.

5.2 Data Collection and Annotation

The images used for training in the project consisted of self-annotated images and existing datasets. For the existing dataset, the images of plastic bottles originated from multiple sources, including the csproject (2024), the WaRP - Waste Recycling Plant Dataset compiled by Yudin et al. (2023), and the Plastic Bottles Dataset by SnapCycleV2 (2023). The images were selectively picked from these datasets. The labelled images were then uploaded to Roboflow, which labelled the plastic bottles in the images based on the labels provided.

For the self-annotated images, they were first uploaded to Roboflow, where bounding box tools were used to annotate the plastic bottles in the images. If there were no plastic bottles in the image, the images were labelled as null to represent the background.

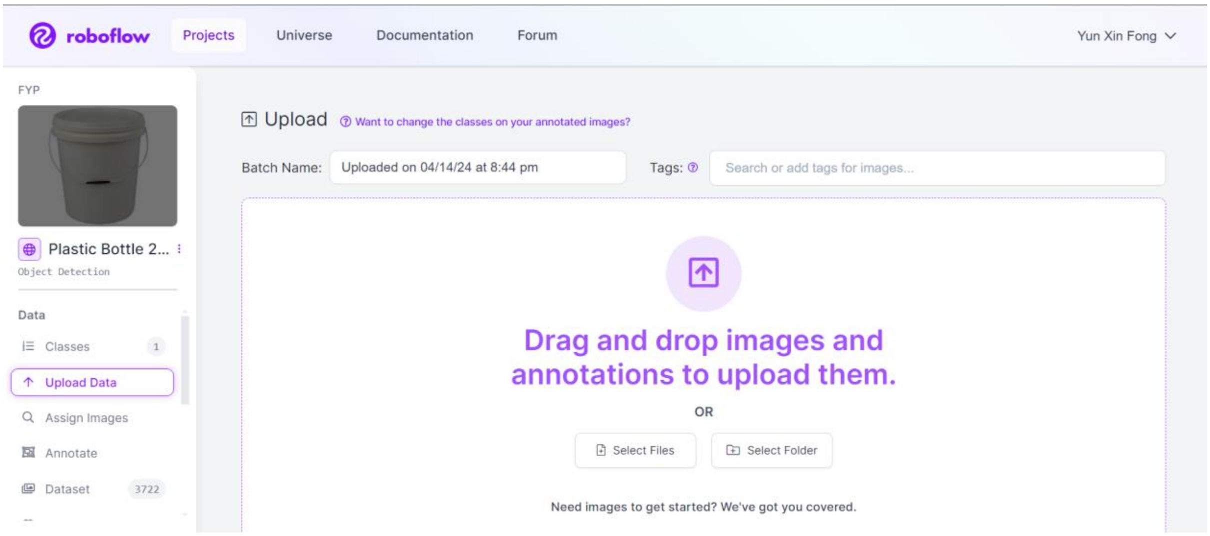


Figure 5.1: Upload data to Roboflow

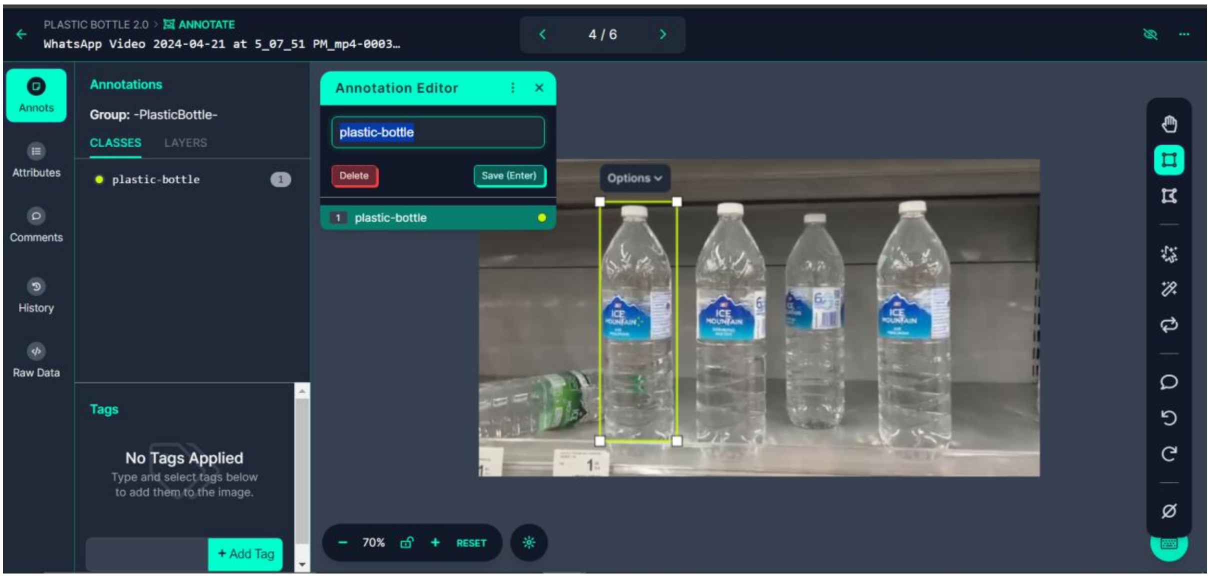


Figure 5.2: Annotate plastic bottle in the image.

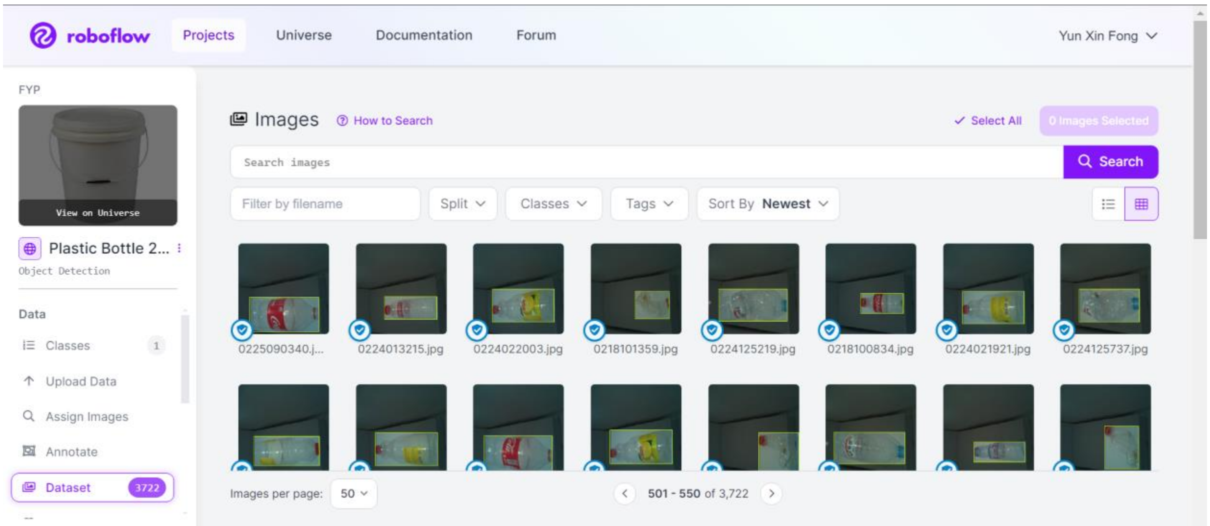


Figure 5.3: Create dataset in Roboflow with label.

5.3 Data Preprocessing and Augmentation

Preprocessing is applied to the images in the dataset, including auto orientation and resizing. Furthermore, data augmentation is also applied to the dataset to increase the number of images for the training set. The augmentations added to the images include flipping (horizontal & vertical), 90° rotation (clockwise, counter-clockwise, and upside down), rotation (between -15° and +15°), shear (±14° horizontal and ±14° vertical), brightness adjustment (between -15% and +15%), blur (up to 1.1 pixels), and noise (up to 0.1% of pixels). These data augmentations are randomly applied to the images in the dataset.

Preprocessing	Auto-Orient: Applied Resize: Stretch to 800x800
Augmentations	Outputs per training example: 3 Flip: Horizontal, Vertical 90° Rotate: Clockwise, Counter-Clockwise Rotation: Between -15° and +15° Shear: ±14° Horizontal, ±14° Vertical Brightness: Between -15% and +15% Blur: Up to 1.1px Noise: Up to 0.1% of pixels

Figure 5.4: Apply data preprocessing and the data augmentation to the datasets.

Figure 5.5: Example of data augmentation in the training sets.

5.4 Data Splitting

The dataset is split to 70%, 20%, and 10% for training, validation, and test dataset. Therefore, a total of 7100 images in the dataset is split into 5004 for training, 1396 for validation, and 700 for testing. The datasets will split in three different folder which is train, val, and test folder which contains the images and the labels. The dataset is exported into different format to be used in different pre-trained deep learning models. The dataset is exported as Pascal VOC, YOLOv8, TFRecord, COCO for different pre-trained deep learning models' training.

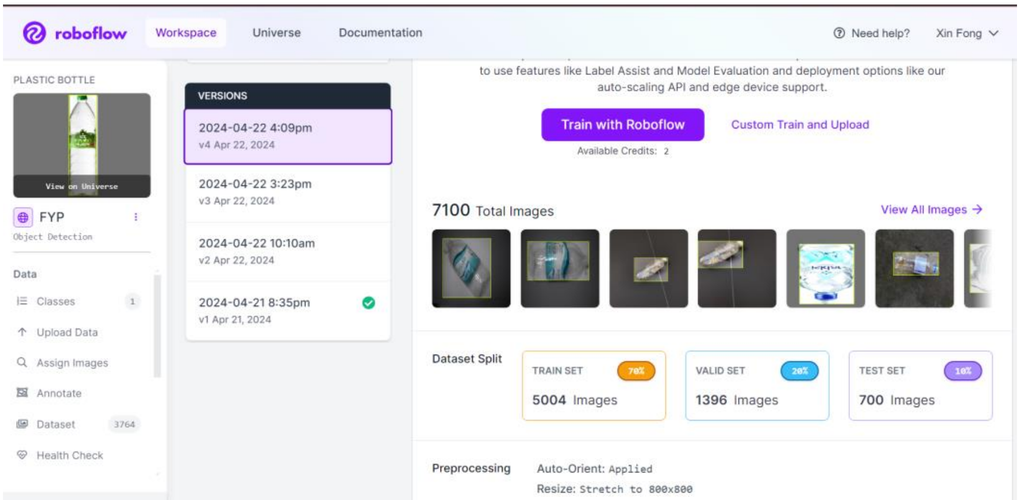


Figure 5.6: Dataset splits into train set, valid set and test set

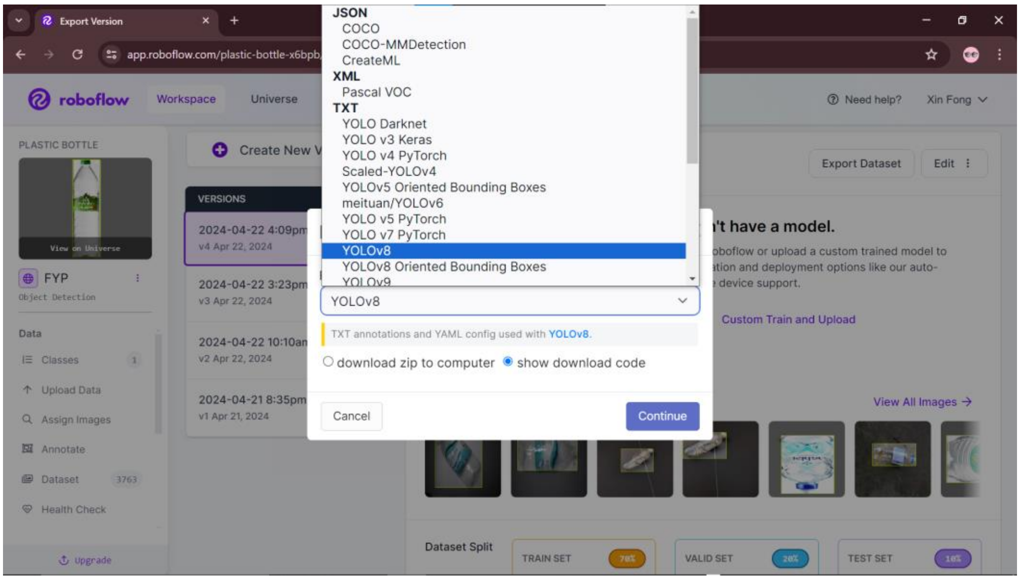


Figure 5.7: Dataset exports with different format

5.5 Deep Learning Models Setup and Training

The training of the deep learning models is done on Kaggle and Google Colab to utilize the free GPU resources in order to increase the training speed. After successfully training the Faster R-CNN R50-FPN, SSD MobileNet-v2, and YOLOv8, the trained deep learning models will be downloaded to evaluate their performance. The training process will be repeated to optimize the performance of the deep learning models. The deep learning models will be trained using different epochs and batch sizes to achieve the optimum performance for PET-plastic bottle detection.

5.6 Deep Learning Models Comparison

The results show that YOLOv8 has the best performance in detecting plastic bottles among all the pre-trained deep learning models. YOLOv8 achieved a mean Average Precision (mAP) of 0.923 on the custom dataset at an intersection over union (IOU) threshold of 0.5, compared to Faster R-CNN and SSD MobileNet v2 which has mAP 0.6544 and 0.643 respectively. Therefore, YOLOv8 is chosen for model deployment and will be used to detect plastic bottles on the moving conveyor belt in recycling centres.

Table 5.1: Deep learning models comparison results

Pre-Trained Deep Learning Models	Faster R-CNN R50-FPN	SSD MobileNet-v2	YOLOv8
Mean Average Precision @mAP at intersection over union (IOU): 0.5	0.6544	0.643	0.923
Mean Average Precision @mAP at intersection over union (IOU): 0.50:0.95:0.05	0.4629	0.294	0.733

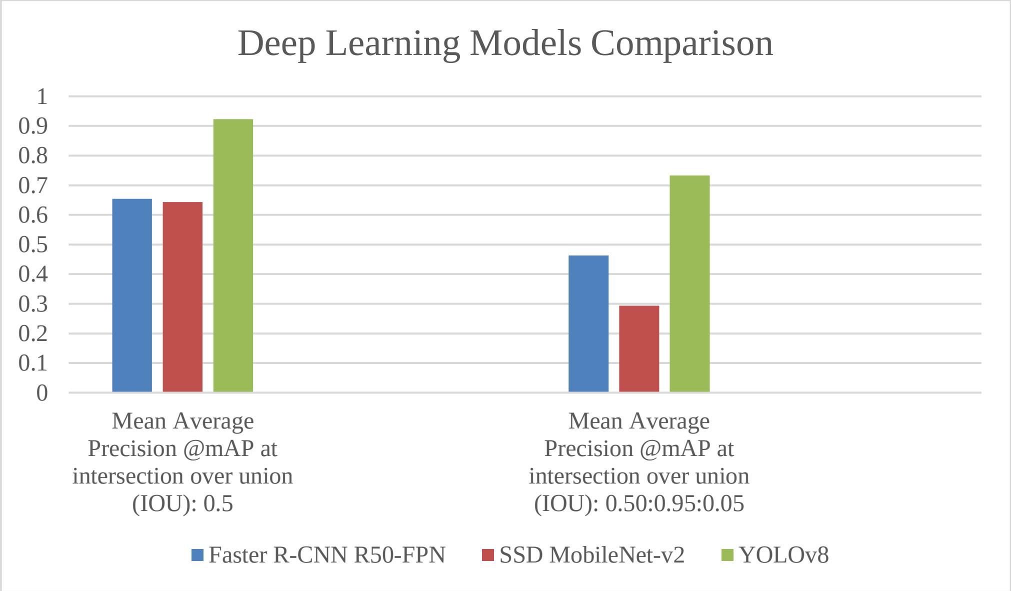


Figure 5.8: Comparison of deep learning models

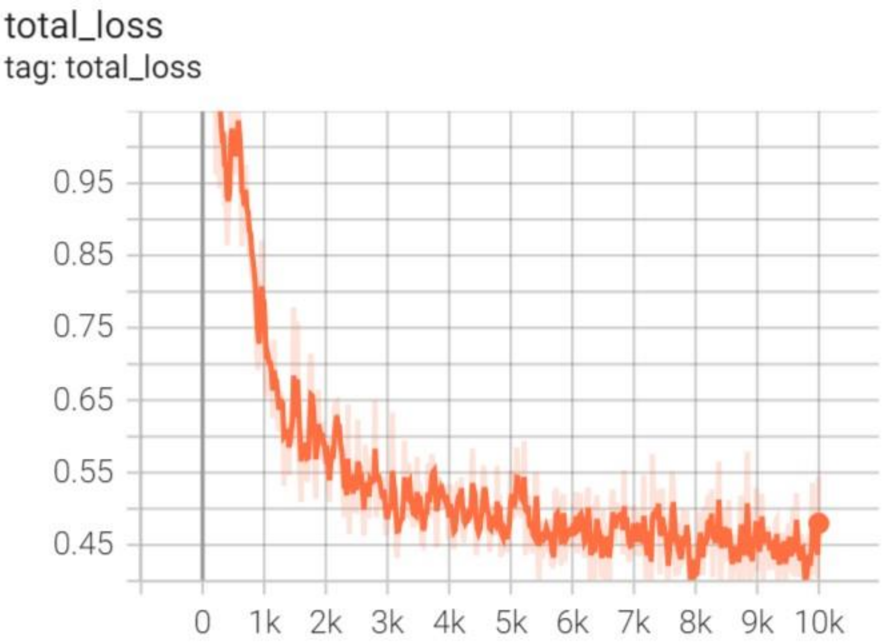


Figure 5.9: Faster R-CNN R50-FPN total loss

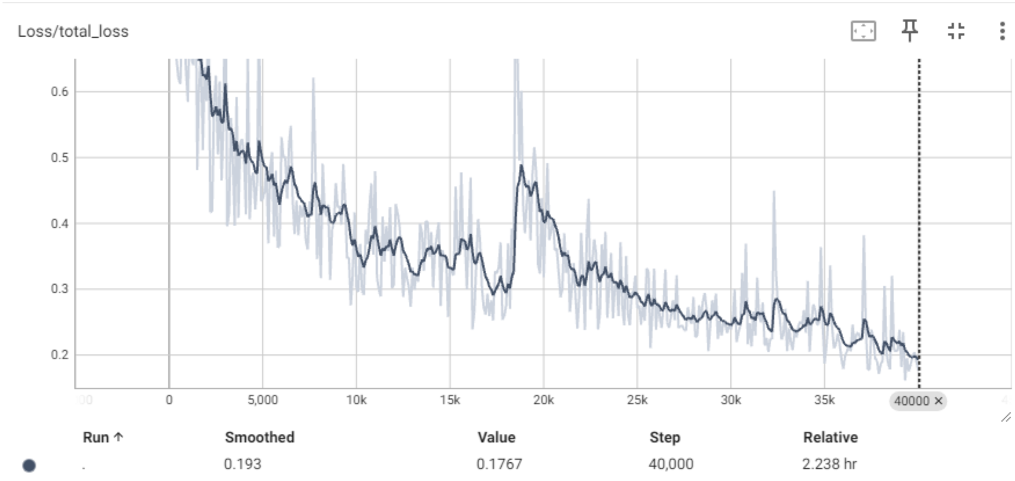


Figure 5.10: SSD MobileNetV2 total loss

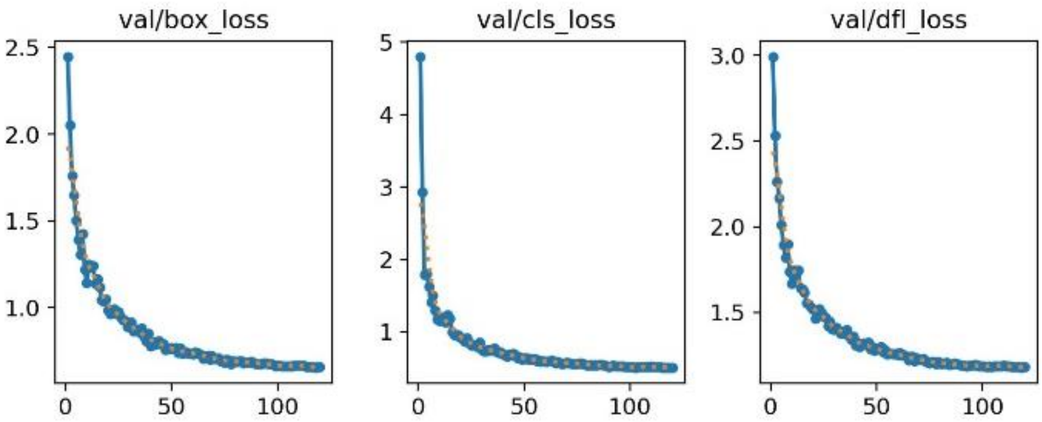


Figure 5.11: YOLOv8 total loss

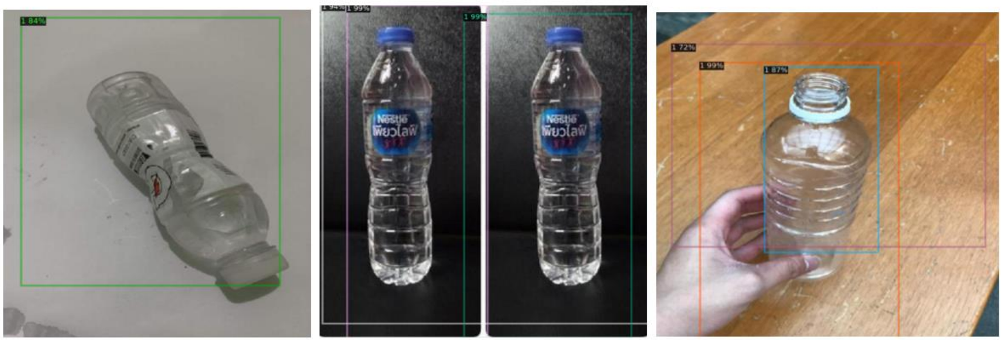


Figure 5.12: Prediction results for Faster R-CNN R50-FPN

Figure 5.13: Prediction results for SSD MobileNetv2

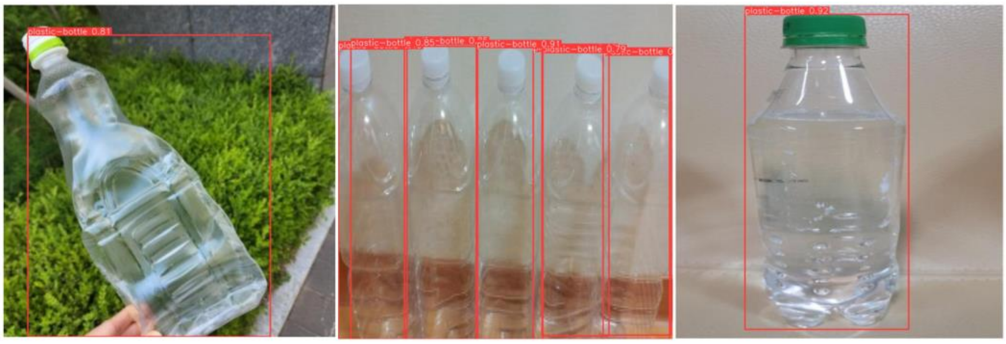


Figure 5.14: Prediction results for YOLOv8

5.7 Model Deployment

As YOLOv8 achieved the highest mean average precision among the pre-trained deep learning models, the selected pre-trained YOLOv8 model is used to predict plastic bottles on the moving conveyor belt in the recycling centre. The average precision for detecting plastic bottles on the moving conveyor belt in the recycling centre is 0.3026, and the average recall is 0.5188.

Table 5.2: Precision and recall for waste products on moving conveyer belt in the recycling centre.

Video	TP	FP	FN	Precision	Recall
01	5	6	2	0.4545	0.7143
02	2	6	4	0.2500	0.3333
03	7	5	4	0.5833	0.6363
04	5	10	3	0.3333	0.6250
05	8	12	3	0.4000	0.7272
06 (No plastic bottle in the video)	0	8	0	0	0
07	1	5	1	0.1667	0.5000
08	4	5	1	0.4444	0.8000
09	1	10	2	0.0909	0.3333

Figure 5.15: Example of YOLOv8 Prediction output on Video 02

Figure 5.16: Example of YOLOv8 Prediction output on Video 05

Figure 5.17: Example of YOLOv8 Prediction output on Video 09

To improve precision in detecting plastic bottles on the moving conveyor belt, the waste products on the conveyer belt are spread more evenly to reduce overlapping. The average precision for the less overlapping waste products on the moving conveyor belt has improved significantly to 0.6783, and there has also been a slight improvement in the average recall, which has increased from 0.5188 to 0.6471.

Table 5.3: Precision and recall for less overlapping waste products on moving conveyer belt in the recycling centre.

Video	TP	FN	FP	Precision	Recall
VID_20240419_113613	1	2	5	0.1667	0.3333
VID_20240419_113818	5	2	2	0.7143	0.7143
VID_20240420_141858	20	5	5	0.8000	0.8000
VID_20240420_142224	11	2	10	0.5238	0.8462
VID_20240420_142504	18	9	8	0.6923	0.6667
VID_20240420_142801	10	15	6	0.6250	0.4000
VID_20240420_142913	10	3	4	0.7143	0.7692

The precision of the detection of PET-plastic bottles on the moving conveyer belt is affected by the similar material of the waste products. The YOLOv8 classify the items as plastic bottles if the items have similar appearance or textures with the PET-plastic bottles. Moreover, the size of the PET-plastic bottles is limit to size ranging from 500 millilitres to 1000 millilitres. Thus, the precision of the detection rate drops when the YOLOv8 predicts that PET-plastic bottles that has greater volume than 1000 millilitres. The limitation of the YOLOv8 will be further discussed in the conclusion.

Figure 5.18: Example of YOLOv8 prediction output on video VID_20240419_113818

Figure 5.19: Example of YOLOv8 prediction output on video VID_20240420_141858

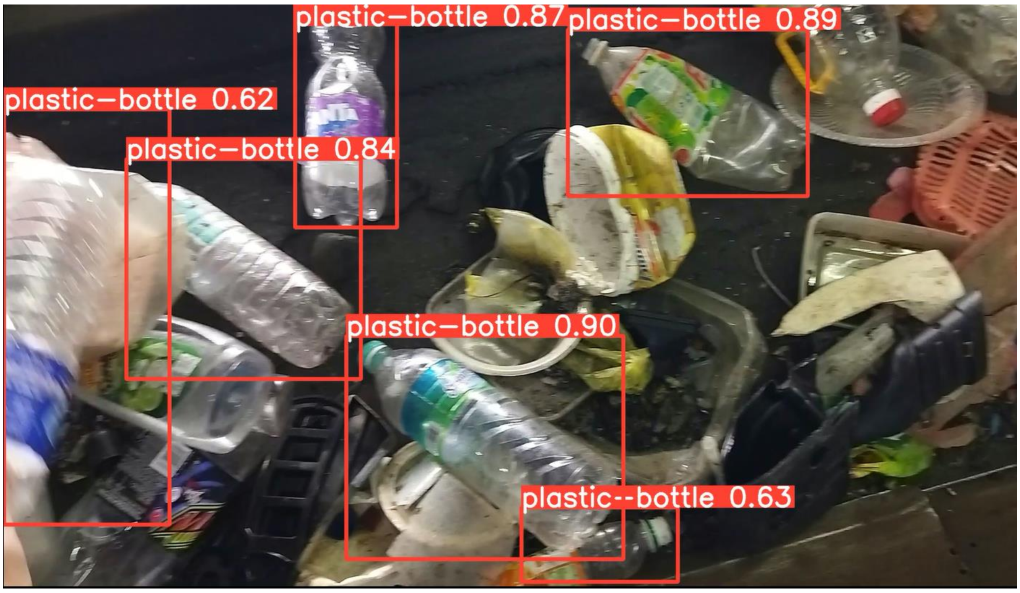


Figure 5.20: Example of YOLOv8 prediction output on video VID_20240420_141858

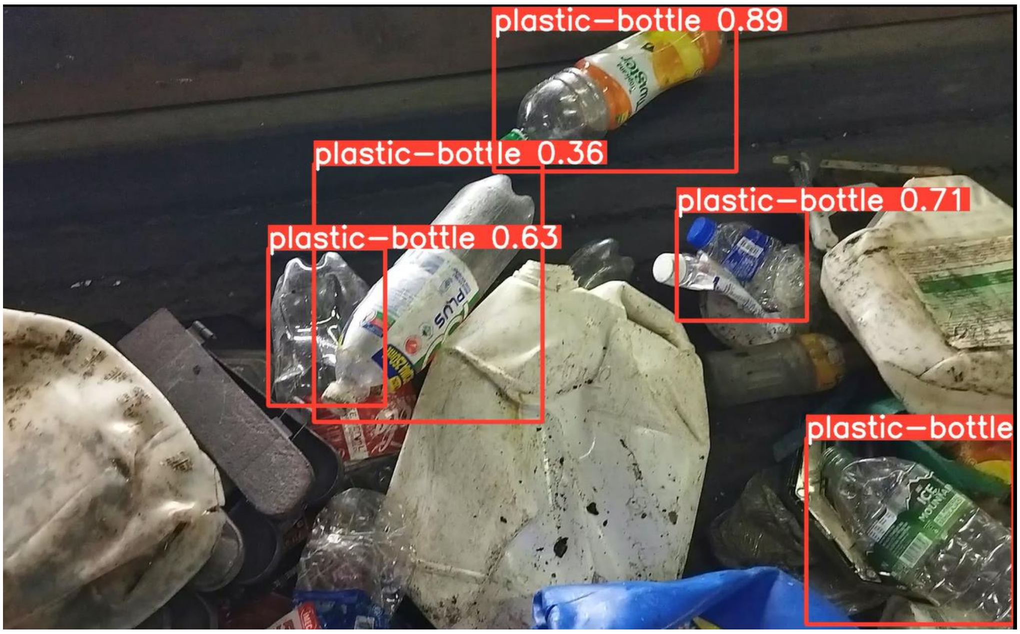


Figure 5.21: Example of YOLOv8 prediction output on video VID_20240420_142913

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 Introduction

In this chapter, the YOLOv8 has been selected is used to implement in the application because YOLOv8 has the best performance compared to Faster R-CNN and SSD. The application is developed to ease the user who do not have the background in the deep learning models to able to use the YOLOv8 model. A simple user interface is developed to integrate with the pre-trained YOLOv8 model.

6.2 User Authentication of the Application

The application is developed using Python and Gradio who allows developers to build an interactive user interface for deep learning model. A simple login function is implemented to make sure that only people have permission able to use the application. User able to view the user interface once login to the application successfully.

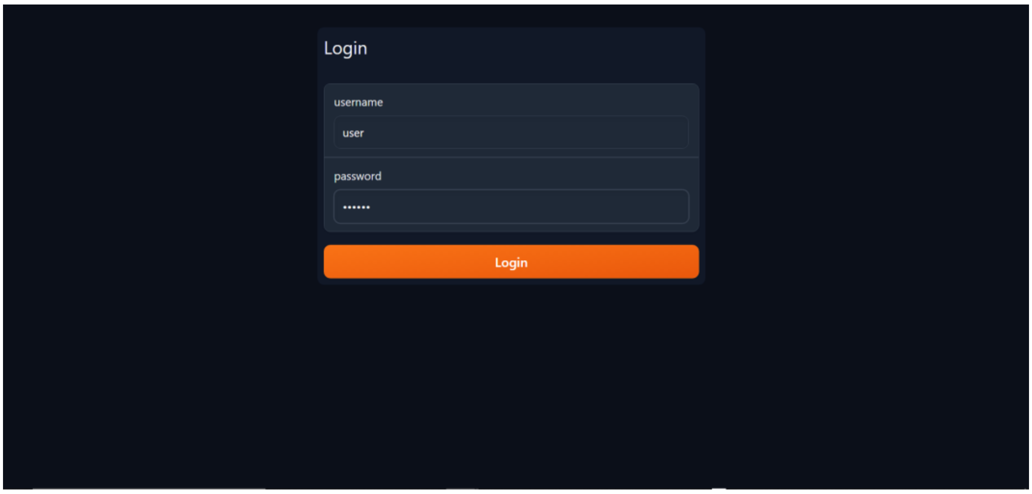


Figure 6.1: Login interface for the application

```
if __name__ == '__main__':
    get_frame_rate()
    input_interface.launch(auth=('[redacted]', '[redacted]'))
```

Figure 6.2: User authentication before login to the application

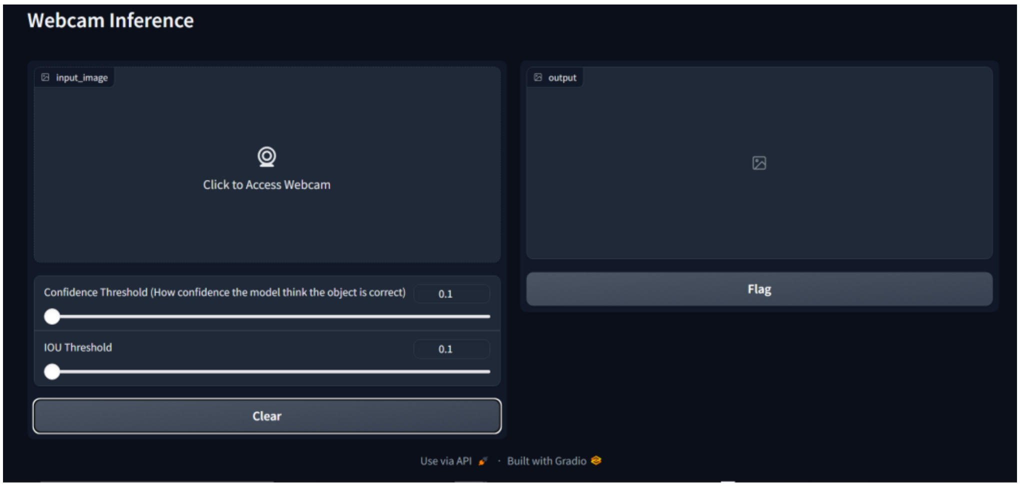


Figure 6.3: User interface for the application after login to the application

6.3 Webcam Inference on the Application

After user login to the application successfully, user clicks the icon on the input to get access to the application. The application will prompt the user to permit the access to the webcam on the user's device.

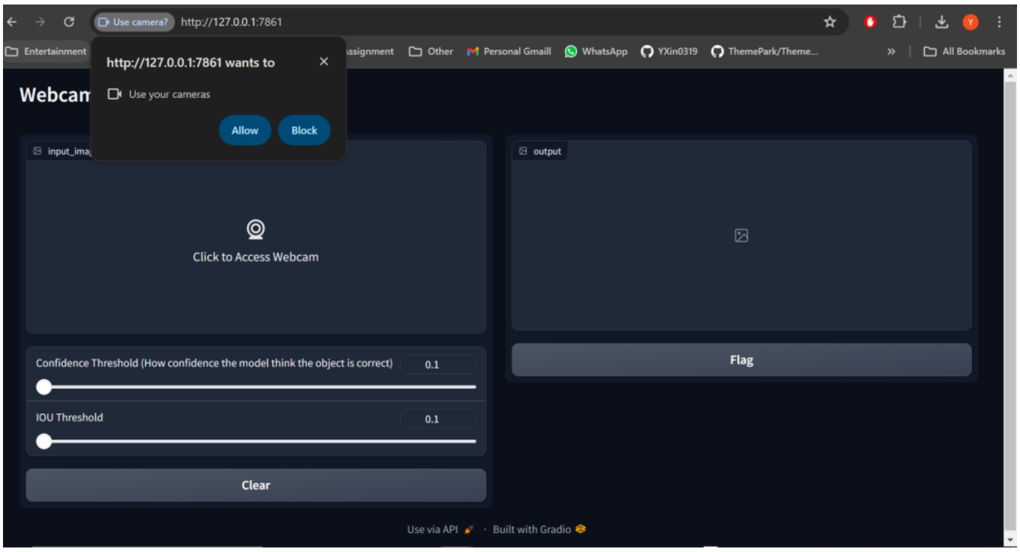


Figure 6.4: Get user permission to access the webcam.

```python
def predict_image(frame, conf_threshold, iou_threshold):
    img = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = model.predict(
        source=img,
        conf=conf_threshold,
        iou=iou_threshold,
        show_labels=True,
        show_conf=True,
        imgsz=640,
    )

    for r in results:
        im_array = r.plot()
        im = Image.fromarray(im_array[..., ::-1])

    return im

def webcam_inference(input_image, conf_threshold=0.25, iou_threshold=0.45):
    frame = np.array(input_image)
    im = predict_image(frame, conf_threshold, iou_threshold)
    return im

input_interface = gr.Interface(
    fn=webcam_inference,
    inputs=[
        gr.Image(sources=["webcam"], streaming=True),
        gr.Slider(minimum=0.1, maximum=1, label="Confidence Threshold (How confident the model thinks the object is correct)"),
        gr.Slider(minimum=0.1, maximum=1, label="IOU Threshold")
    ],
    outputs="image",
    title="Webcam Inference",
    live=True
)
```

Figure 6.5: Code segment for the user interface of the application.

Once the user starts the webcam, they can wait for the output to show the prediction on the webcam input for the plastic bottles. In the command prompt, the speed for inferring a frame from the webcam can also be calculated and show the number of plastic bottles detected in that frame.

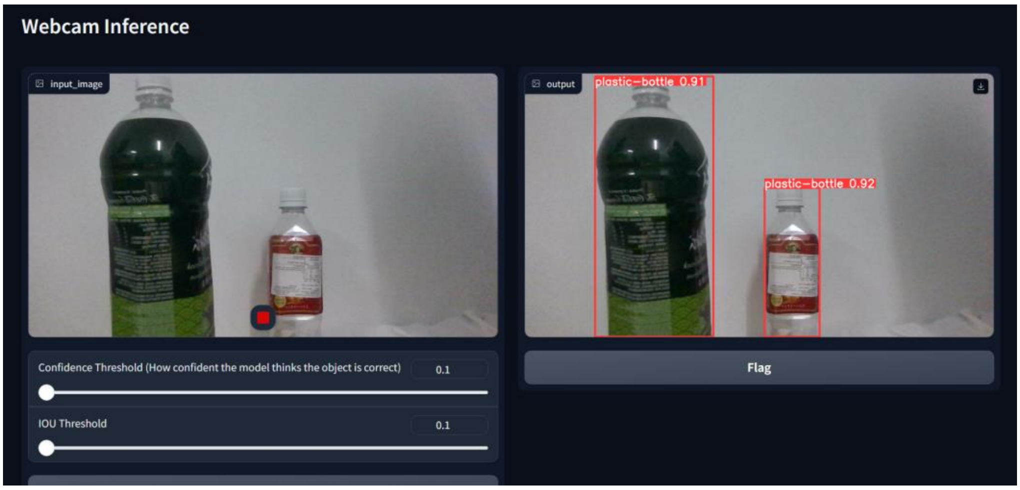


Figure 6.6: Start the webcam for plastic bottles prediction.

```
0: 384x640 (no detections), 468.3ms
Speed: 21.0ms preprocess, 468.3ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 2 plastic-bottles, 250.0ms
Speed: 5.0ms preprocess, 250.0ms inference, 6.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 plastic-bottle, 231.0ms
Speed: 4.0ms preprocess, 231.0ms inference, 2.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 plastic-bottle, 252.0ms
Speed: 5.0ms preprocess, 252.0ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 279.3ms
Speed: 7.0ms preprocess, 279.3ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 215.0ms
Speed: 5.0ms preprocess, 215.0ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 plastic-bottle, 228.0ms
Speed: 5.0ms preprocess, 228.0ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 223.9ms
Speed: 5.0ms preprocess, 223.9ms inference, 2.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 215.0ms
Speed: 3.0ms preprocess, 215.0ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 251.7ms
Speed: 4.0ms preprocess, 251.7ms inference, 0.0ms postprocess per image at shape (1, 3, 384, 640)
```

Figure 6.7: The number of plastic bottles detected and speed to complete prediction on one frame.

6.4 Conclusion

The user interface and the YOLOv8 has now successfully integrated. Unit testing, integrating testing, usability testing and user acceptance testing will be conducted once the application is successfully developed.

CHAPTER 7

SYSTEM TESTING

7.1 Introduction

This chapter includes all testing for the application to make sure it fulfils the functional requirements and the non-functional requirements of the project. In this chapter, the testing covered for the project are unit testing, integration testing, usability testing and user acceptance testing.

7.2 Unit Testing

Unit testing is conducted to make sure all the function in the application is able to work without any issues.

Table 7.1: Test case for Login module

Project Name:	Application Development for Plastic Bottle Detection using Deep Learning				**Test Designed by:**	Fong Yun Xin		
Module Name:	Login Module				**Test Designed date:**	23/04/2024		
Release Version:	Version 1.0				**Test Executed by:**	Fong Yun Xin		
					Test Execution date:	23/04/2024		
Test Priority	Low							
Test Case#	**Test Title**	**Test Summary**	**Test Steps**	**Test Data**	**Expected Result**	**Post-condition**	**Actual Result**	**Status**
Test_Login_01	Test Verify Rule 1	If user clicks	1. User enters wrong username 2. User enters wrong password	Username: User123 Password:1234567	Invalid credential	User needs to re-enter password	Invalid credential	Pass
Test_Login_02	Test Verify Rule 2	If user enters correct username and wrong password	1. User enters correct username 2. User enters wrong password	Username: User Password:1234568	Invalid credential	User needs to re-enter password	Invalid credential	Pass

Test_Login_03	Test Verify Rule 3	If user enters correct username and wrong password	1. User enters wrong username 2. User enters correct password	Username: User123 Password: abc123	Invalid credential	User needs to re-enter password	Invalid credential	Pass
Test_Login_04	Test Verify Rule 4	If user enters correct username and wrong password	1. User enters correct username 2. User enters correct password	Username: User Password: abc123	User successfully Login	Login successfully	User successfully Login	Pass

Table 7.2: Decision table for Login module

Conditions	Test Verify Rule 1	Test Verify Rule 2	Test Verify Rule 3	Test Verify Rule 4
Username	Invalid	Valid	Invalid	Valid
Password	Invalid	Invalid	Valid	Valid
Actions				
Login Succeed	False	False	False	True

Table 7.3: Test case for Start webcam module

Project Name:	Application Development for Plastic Bottle Detection using Deep Learning				**Test Designed by:**	Fong Yun Xin		
Module Name:	Start Webcam Module				**Test Designed date:**	24-04-2024		
Release Version:	Version 1.0				**Test Executed by:**	Fong Yun Xin		
					Test Execution date:	24-04-2024		
Pre-condition	User successfully login to the application							
Test Priority	High							
Test Case#	**Test Title**	**Test Summary**	**Test Steps**	**Test Data**	**Expected Result**	**Post-condition**	**Actual Result**	**Status**
Test_Webcam_01	Test Verify Rule 5	If user clicks open webcam and allows application access the webcam	1. User clicks the webcam 2. User clicks allow application to access webcam		Webcam successfully open.		Webcam successfully open.	Pass

Test_Webcam_02	Test Verify Rule 6	If user clicks do not open webcam and blocks application access the webcam	1. User clicks the webcam 2. User clicks block application to access webcam		Webcam does not open.		Webcam does not open.	Pass
Test_Webcam_03	Test Verify Rule 7	If user clicks do not open webcam and blocks application access the webcam	1. User does not click the webcam		Webcam does not open.		Webcam does not open.	Pass

Table 7.3: Decision table for Start webcam module

Conditions	Test Verify Rule 5	Test Verify Rule 6	Test Verify Rule 7
Open Webcam	True	True	False
Allow Webcam Access	True	False	False
Actions			
Webcam	Open	Close	Close

7.3 Integration Testing

In integration testing, the start webcam module and the show output module are tested together to test the detection on the plastic bottle application to make sure that the output able to produce prediction.

Table 7.4: Test case for plastic bottle detection integrating testing

Project Name:	Application Development for Plastic Bottle Detection using Deep Learning				**Test Designed by:**	Fong Yun Xin		
Test Type:	Integration Testing				**Test Designed date:**	24-04-24		
Release Version:	Version 1.0				**Test Executed by:**	Fong Yun Xin		
					Test Execution date:	24-04-2024		
Pre-condition	User successfully login to the application							
Test Priority	High							
Test Case#	**Test Title**	**Test Summary**	**Test Steps**	**Test Data**	**Expected Result**	**Post-condition**	**Actual Result**	**Status**

Test_Detection_01	Test Verify Rule 8	If user clicks open webcam and allows application access the webcam	1. User clicks the webcam 2. User clicks allow application to access webcam		Webcam successfully open and output show prediction. Command prompt shows prediction output		Webcam successfully open and output show prediction	Pass
Test_Detection_02	Test Verify Rule 9	If user clicks do not open webcam and blocks application access the webcam	1. User clicks the webcam 2. User clicks block application to access webcam		Webcam does not open and no output shows		Webcam does not open and no output shows	Pass

Table 7.5: Decision table for plastic bottle detection integrating testing

Conditions	Test Verify Rule 8	Test Verify Rule 9
Open Webcam	True	True
Allow Webcam Access	True	False
Actions		
Webcam	Open	Close
Output	True	False
Prediction Log	True	False

7.4 Usability Testing

To make sure that the application is able to work and easy for user who does not have deep learning knowledge to use, the usability test is conducted to evaluate the application. The testers who involved in the testing consists of students who have deep learning knowledge backgrounds and students who does not have deep learning knowledge backgrounds.

All the testers are required to fill in the user satisfaction survey after they test the application. The user satisfactory survey and the consent form are attached as "Appendix B: User Satisfactory Survey". The following are the results for the user satisfaction survey.

Table 7.6: Results for the user satisfaction survey.

Participant #	Score by Question #										SUS Score
	1	2	3	4	5	6	7	8	9	10	
1	5	1	5	2	5	2	5	1	5	1	95
2	5	1	5	1	4	1	5	1	5	1	97.5
3	5	2	5	2	4	1	4	1	4	2	85
4	4	3	4	2	4	3	4	2	4	2	70
5	4	1	5	4	4	2	4	2	3	2	72.5
Average	4.6	1.6	4.8	2.2	4.2	1.8	4.4	1.4	4.2	1.6	84

Participant 1 and participant 2 are student who has knowledge in deep learning while participant 3, 4 and 5 are student who does not have knowledge. Student who has does not knowledge in deep learning may find it a bit hard to use the application compared with students who has knowledge in deep learning. The average system usability scale (SUS) score is 84 which means the application usability performance is excellent.

For the comments received from the participants, the application should consider detecting more types of waste products for future improvements and also have faster detection rate for the real time plastic bottle detection.

7.5 User Acceptance Testing

The user acceptance testing is conducted to test the application to make sure that the application able to work as expected. The participants are given the scenario to perform the desired action in the application. The moderators will observe whether the participants are able to carry out desired action or not. The details of the user acceptance testing results are included in the appendix as “Appendix C: User Acceptance Testing Results”.

Table 7.6: Results for the user satisfaction survey.

Participant #	Test Modules				Comments
	Login	Start Webcam	Stop Webcam	Show Prediction Output	
1	Pass	Pass	Pass	Pass	No
2	Pass	Pass	Pass	Pass	No
3	Pass	Pass	Pass	Pass	No
4	Pass	Pass	Pass	Pass	The prediction output is a bit slow.
5	Pass	Pass	Pass	Pass	No

CHAPTER 8

CONCLUSION AND FUTURE WORKS

In this project, three pre-trained deep learning models, namely Faster R-CNN, SSD, and YOLOv8, were selected to train using PET-plastic bottle images. The model performance for all three pre-trained deep learning models was compared using mean average precision (mAP), and YOLOv8 achieved the highest mean average precision of 0.923 among all the pre-trained deep learning models. The results of the pre-trained deep learning models demonstrate that it is a feasible solution to replace manual labour in sorting plastic bottles in recycling centres and increase the efficiency of the sorting process.

YOLOv8 was selected and used to predict plastic bottles on the moving conveyor belt in the recycling centre. Although the average precision and recall are not as high on the custom dataset, YOLOv8 still proves to be a good solution for detecting PET-plastic bottles. YOLOv8 was deployed with a simple user interface for users without a background in deep learning to use. The application passed usability testing, and user acceptance tests showed that users without deep learning knowledge were able to use the application without any problems.

For future work, the best-performing pre-trained deep learning model will be passed to other engineering faculty departments at Universiti Tunku Abdul Rahman to integrate with robotic arms to assist in sorting plastic bottles in recycling centres. Additionally, more categories of plastic items will be added to the dataset to enable the model to classify them. Furthermore, additional categories of waste products such as paper, glass, metal, etc., could also be added to the datasets for future training. Moreover, larger variations of YOLOv8 such as YOLOv8m, YOLOv8l, etc., could also be trained to assess improvements in precision in detecting plastic bottles. Nevertheless, the project still has the limitation and rooms for improvement in future. In the following table is the details of the limitation and the recommendations for improvement.

Table 8.1: Limitation of the project

No.	Limitation	Reason	Recommendation
1	The speed of application prediction rate (FPS) is slow and have latency to produce output of the prediction results	The application runs on CPU.	Use GPU to speed up the prediction for the output and reduce the latency to show the output.
2	The application only detects PET-plastic bottles.	The deep learning models is trained with PET-plastic bottles ranging from the size 500 milliliters to 1000 milliliters only.	Add more types of plastic bottles and other waste categories to the dataset.
3	The application has high false positive (FP) when detecting plastic bottles on the moving conveyer belt in the video.	Many waste products in the video have a similar appearance or texture to PET-plastic bottles. One common misdetection for YOLOv8 is predicting PET-plastic bottles that are larger in size (more than 1000 millilitre) as positive because they are PET-plastic and differ only in size.	Annotate the plastic bottles based on size and material for better detection rate and easier for classification in future works.

4	The performance of the deep learning models still does not reach the optimum, and the accuracy of the model detection should continue to improve.	Limited computational resources provided for this project.	Get more computational resources by buying a GPU or subscribing to online GPU cloud services like Microsoft Azure or Pay for Google Colab for more GPU quota.
5	The precision on detecting the overlapping of the plastic bottles’ wastes on the moving conveyer belt is lower than the precision on detecting the less overlapping of the plastic bottles’ wastes on the moving conveyer belt.	The overlapping of the plastic bottles' waste blocks some parts of the bottles.	Reduce the overlapping of plastic bottle waste for better detection and achieve higher precision.
6	Confidence threshold of the prediction results is set to default 0.25.	All the prediction output is based on the confidence threshold 0.25.	Try to predict with different confidence threshold and the most suitable confidence threshold for plastic bottle detection.

REFERENCES

Ahmad, M.O., Markkula, J., & Oivo, M. (2013). Kanban in software development: A systematic literature review. 2013 39th Euromicro Conference on Software Engineering and Advanced Applications [Preprint]. doi:10.1109/seaa.2013.28.

Akbar, F.S.P., Ginting, S.Y.P., Wu, G.C., Achmad, S., & Sutoyo, R. (2022, October). Object Detection on Bottles Using the YOLO Algorithm. In 2022 4th International Conference on Cybernetics and Intelligent System (ICORIS) (pp. 1-5). IEEE.

Anderson, D.J., 2010. Kanban: successful evolutionary change for your technology business. Blue Hole Press.

Arai, Y., Miyagusuku, R., & Ozaki, K. (2021, January). Development and Testing of Garbage Detection for Autonomous Robots in Outdoor Environments. In 2021 IEEE/SICE International Symposium on System Integration (SII) (pp. 138-143). IEEE.

Armitage, S., Awty-Carroll, K., Clewley, D., & Martinez-Vicente, V. (2022). Detection and Classification of Floating Plastic Litter Using a Vessel-Mounted Video Camera and Deep Learning. Remote Sensing, 14(14), p.3425.

Boesch, G., 2023. *Object Detection in 2023: The Definitive Guide.* [Online] Available at: https://viso.ai/deep-learning/object-detection/ [Accessed 2 September 2023].

Boesch, G., 2024. *A Guide to YOLOv8 in 2024.* [Online] Available at: https://viso.ai/deep-learning/yolov8-guide/ [Accessed 20 April 2024].

Bochkovskiy, A., Wang, C.Y., & Liao, H.Y.M. (2020). Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

Chen, H.L. et al. (2021). The Plastic Waste Problem in Malaysia: Management, recycling and disposal of local and Global Plastic Waste. SN Applied Sciences, 3(4). doi:10.1007/s42452-021-04234-y.

Chen, K., An, J., Fang, Y., & Bu, T. (2022, July). Research on solid waste plastic bottle cognitive based on YOLOv5s and deep stochastic configuration network. In 2022 7th International Conference on Automation, Control and Robotics Engineering (CACRE) (pp. 275-280). IEEE.

csproject. (2024). Object detection Dataset [Open Source Dataset]. Roboflow Universe. Roboflow. Retrieved from https://universe.roboflow.com/csproject-lqqy2/object-detection-mfzpy [Accessed 11 Apr. 2024].

Do, H.T. and Thi, L.P., 2020. Artificial intelligence (AI) application on plastic Kim, M. and Kim, S., 2021, October. YOLO-based robotic grasping. In 2021 21st International Conference on Control, Automation and Systems (ICCAS) (pp. 1120-1122). IEEE.bottle monitoring in coastal zone. J. Hydrometeorol, 6, pp.57-67.

Dmitry Yudin, Nikita Zakharenko, Artem Smetanin, Roman Filonov, Margarita Kichik, Vladislav Kuznetsov, Dmitry Larichev, Evgeny Gudov, Aleksandr Panov, and Semen Budennyy. (2023). WaRP - Waste Recycling Plant Dataset [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DS/3027350.

Fulton, M., Hong, J., Islam, M.J., & Sattar, J. (2019, May). Robotic detection of marine litter using deep visual detection models. In 2019 international conference on robotics and automation (ICRA) (pp. 5752-5758). IEEE.

Gad, A. F., 2020. *Faster R-CNN Explained for Object Detection Tasks.* [Online] Available at: https://blog.paperspace.com/faster-r-cnn-explained-object-

detection/
[Accessed 2 September 2023].

Hasany, S.N., Zaidi, S.S., Sohail, S.A. and Farhan, M., 2021, July. An autonomous robotic system for collecting garbage over small water bodies. In 2021 6th International Conference on Automation, Control and Robotics Engineering (CACRE) (pp. 81-86). IEEE.

Hui, J., 2018. *SSD object detection: Single Shot MultiBox Detector for real-time processing.* [Online]
Available at: https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06

IBM Data and AI Team. (2023). AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the difference?. Available at: URL [Accessed 13 August 2023].

Kim, M. and Kim, S., 2021, October. YOLO-based robotic grasping. In 2021 21st International Conference on Control, Automation and Systems (ICCAS) (pp. 1120-1122). IEEE.

Kondu, R., 2023. *YOLO: Algorithm for Object Detection Explained [+Examples].* [Online]
Available at: https://www.v7labs.com/blog/yolo-object-detection#:~:text=Key%20takeaways-,YOLO%20(You%20Only%20Look%20Once)%20is%20a%20popular%20object%20detection,real%2Dtime%20object%20detection%20tasks.
[Accessed 20 August 2023].

Ladas, C., 2009. *Scrumban-essays on kanban systems for lean software development*. Lulu. com.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., & Li, Y. (2022). YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.

Lin, F., Hou, T., Jin, Q., & You, A. (2021). Improved YOLO based detection algorithm for floating debris in waterway. Entropy, 23(9), p.1111.

Lin, W., 2021, December. YOLO-green: A real-time classification and object detection model optimized for waste management. In 2021 IEEE International Conference on Big Data (Big Data) (pp. 51-57). IEEE.

Liu, L., Zhou, B., Liu, G., Lian, D., & Zhang, R. (2022, July). Yolo-Based Multi-Model Ensemble for Plastic Waste Detection Along Railway Lines. In IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium (pp. 7658-7661). IEEE.

Mao, W.L., Chen, W.C., Fathurrahman, H.I.K., & Lin, Y.H. (2022). Deep learning networks for real-time regional domestic waste detection. Journal of Cleaner Production, 344, p.131096.

O'Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G.V., Krpalkova, L., Riordan, D., & Walsh, J. (2020). Deep learning vs. traditional computer vision. In Advances in Computer Vision: Proceedings of the 2019 Computer Vision Conference (CVC), Volume 1 (pp. 128-144). Springer International Publishing.

Poppendieck, M., & Cusumano, M.A. (2012). Lean software development: A tutorial. IEEE software, 29(5), pp.26-32.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

SnapCycleV2. (2023). Plastic Bottles Dataset [Open Source Dataset]. Roboflow Universe. Roboflow. Retrieved from https://universe.roboflow.com/snapcyclev2/plastic-bottles-ip5yb [Accessed 18 Apr. 2024].

Statista Research Department and 27, J. (2023). Malaysia: Plastic waste trade volume, Statista. Available at: URL [Accessed 07 August 2023].

Tan, M.H., Chiong, M.S., Chun, Y.Y., Tsukahara, K. and Tahara, K., 2022. An Analysis of Practices and Challenges for Plastic Recycling Industry in Malaysia. *International Journal of Automation Technology*, *16*(6), pp.831-837.

Tan, M., Pang, R., & Le, Q.V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).

Tamin, O., Moung, E.G., Dargham, J.A., Yahya, F., Omatu, S., & Angeline, L. (2022, September). A comparison of RGB and RGNIR color spaces for plastic waste detection using the YOLOv5 architecture. In 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) (pp. 1-6). IEEE.

Tornero, P., Puente, S. and Gil, P., 2022, April. Detection and location of domestic waste for planning its collection using an autonomous robot. In 2022 8th International Conference on Control, Automation and Robotics (ICCAR) (pp. 138-144). IEEE.

Wahyutama, A.B., & Hwang, M. (2022). YOLO-based object detection for separate collection of recyclables and capacity monitoring of trash bins. Electronics, 11(9), p.1323.

What is Kanban? explained in 10 minutes: Kanbanize (no date). Kanban Software for Agile Project Management. Available at: URL [Accessed 14 August 2023].

Xiao, J., Tang, Y., Zhao, Y., & Yan, Y. (2020, December). Design of plastic bottle image recognition system based on improved YOLOv3. In 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (pp. 2047-2050). IEEE.

Yang, G., Jin, J., Lei, Q., Wang, Y., Zhou, J., Sun, Z., Li, X., & Wang, W. (2021, October). Garbage classification system with yolov5 based on image recognition. In 2021 IEEE 6th International Conference on Signal and Image Processing (ICSIP) (pp. 11-18). IEEE.

Zailan, N.A., Khairuddin, A.S.M., Khairuddin, U., & Taguchi, A. (2021, June). YOLO-based Network Fusion for Riverine Floating Debris Monitoring System. In 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE) (pp. 1-5). IEEE.

Zhang, Q., Yang, Q., Zhang, X., Wei, W., Bao, Q., Su, J., & Liu, X. (2022). A multi-label waste detection model based on transfer learning. Resources, Conservation and Recycling, 181, p.106235.

Zhou, Q., Liu, H., Qiu, Y., & Zheng, W. (2022). Object Detection for Construction Waste Based on an Improved YOLOv5 Model. Sustainability, 15(1), p.681.

APPENDICES

Appendix A: Gantt Chart

Appendix B: User Satisfactory Survey

No.	Title	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
		1	2	3	4	5
1	I think that I would like to use the application to detect the plastic bottles.					/
2	I found that the application is unnecessarily complex	/				
3	I think the application was easy to use.					/
4	I think the I need technical support to use the application.		/			
5	I find that various function in the application is well integrated.					/
6	I thought there was too much inconsistency in this application.		/			
7	I can imagine that most					/

	people would learn to use this application very quickly.					
8	I found the application very awkward to use.	/				
9	I felt very confident using the application.					/
10	I needed to learn a lot of things before I could get going with this application.	/				

1. What did you like best about the application?
   The accurate of the accuracy rate of plastic bottle.
2. What did you like least about the application?
   Nope
3. Do you have any more comments for the application?
   Nope

Consent &Recording Release Form

I agree to participate in the study conducted and recorded by the Fong Yun Xin, FYP II.

I understand and consent to the use and release of the recording by Fong Yun Xin. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by Fong Yun Xin without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 24-April-2024

Please write your name:___Chong Kae Yi_____________________________

Please sign your name: ____________Kaeyi__________________________

Thank you!

We appreciate your participation.

No.	Title	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
		1	2	3	4	5
1	I think that I would like to use the application to detect the plastic bottles.					/
2	I found that the application is unnecessarily complex	/				
3	I think the application was easy to use.					/
4	I think the I need technical support to use the application.	/				
5	I find that various function in the application is well integrated.				/	
6	I thought there was too much inconsistency in this application.	/				
7	I can imagine that most					/

	people would learn to use this application very quickly.					
8	I found the application very awkward to use.	/				
9	I felt very confident using the application.					/
10	I needed to learn a lot of things before I could get going with this application.	/				

1. What did you like best about the application?
   The application is nice and works fine on detecting the plastic bottles.
2. What did you like least about the application?
   Nope
3. Do you have any more comments for the application?
   Nope

Consent &Recording Release Form

I agree to participate in the study conducted and recorded by the Fong Yun Xin, FYP II.

I understand and consent to the use and release of the recording by Fong Yun Xin. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by Fong Yun Xin without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 24/4/2024

Please write your name: ____Chuah Hui Wen_________________________

Please sign your name: _____huiwen________________________________

Thank you!

We appreciate your participation.

No.	Title	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
		1	2	3	4	5
1	I think that I would like to use the application to detect the plastic bottles.					/
2	I found that the application is unnecessarily complex		/			
3	I think the application was easy to use.					/
4	I think the I need technical support to use the application.		/			
5	I find that various function in the application is well integrated.				/	
6	I thought there was too much inconsistency in this application.	/				
7	I can imagine that most				/	

	people would learn to use this application very quickly.					
8	I found the application very awkward to use.	/				
9	I felt very confident using the application.				/	
10	I needed to learn a lot of things before I could get going with this application.		/			

1. What did you like best about the application?
   User-friendly, easy to use even without knowledge in deep learning.
2. What did you like least about the application?
   Hope It can detect other waste product.
3. Do you have any more comments for the application?
   No

Consent &Recording Release Form

I agree to participate in the study conducted and recorded by the Fong Yun Xin, FYP II.

I understand and consent to the use and release of the recording by Fong Yun Xin. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by Fong Yun Xin without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 24/4/2024

Please write your name: _____Foo Jia Qi___________________________

Please sign your name: _______ _________________________

Thank you!

We appreciate your participation.

No.	Title	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
		1	2	3	4	5
1	I think that I would like to use the application to detect the plastic bottles.				/	
2	I found that the application is unnecessarily complex			/		
3	I think the application was easy to use.				/	
4	I think the I need technical support to use the application.		/			
5	I find that various function in the application is well integrated.				/	
6	I thought there was too much inconsistency in this application.			/		
7	I can imagine that most				/	

	people would learn to use this application very quickly.					
8	I found the application very awkward to use.		/			
9	I felt very confident using the application.				/	
10	I needed to learn a lot of things before I could get going with this application.		/			

1. What did you like best about the application?
   Able to detect plastic bottle through webcam
2. What did you like least about the application?
   Prediction output could be faster
3. Do you have any more comments for the application?
   No

Consent &Recording Release Form

I agree to participate in the study conducted and recorded by the Fong Yun Xin, FYP II.

I understand and consent to the use and release of the recording by Fong Yun Xin. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by Fong Yun Xin without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 25 / 4 / 2024

Please write your name: Fong Yun Qin

Please sign your name:

Thank you!

We appreciate your participation.

No.	Title	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
		1	2	3	4	5
1	I think that I would like to use the application to detect the plastic bottles.				/	
2	I found that the application is unnecessarily complex	/				
3	I think the application was easy to use.					/
4	I think the I need technical support to use the application.				/	
5	I find that various function in the application is well integrated.				/	
6	I thought there was too much inconsistency in this application.		/			
7	I can imagine that most				/	

	people would learn to use this application very quickly.					
8	I found the application very awkward to use.		/			
9	I felt very confident using the application.			/		
10	I needed to learn a lot of things before I could get going with this application.		/			

1. What did you like best about the application?
   Easy to use.
2. What did you like least about the application?
   None.
3. Do you have any more comments for the application?
   Nope

Consent &Recording Release Form

I agree to participate in the study conducted and recorded by the Fong Yun Xin, FYP II.

I understand and consent to the use and release of the recording by Fong Yun Xin. I understand that the information and recording is for research purposes only and that my name and image will not be used for any other purpose. I relinquish any rights to the recording and understand the recording may be copied and used by Fong Yun Xin without further permission.

I understand that participation in this usability study is voluntary and I agree to immediately raise any concerns or areas of discomfort during the session with the study administrator.

Please sign below to indicate that you have read and you understand the information on this form and that any questions you might have about the session have been answered.

Date: 25-April-2024

Please write your name: __Clarisse___________________________________

Please sign your name: _______Clarisse_________________________________

Thank you!

We appreciate your participation.

Appendix C: User Acceptance Testing Results

Participant #	1			
Testing Date	24/04/2024			
UAT ID	Modules	Test Scenario	Results	Comments
UAT_1	Login	1. Participant tries to login the the application with invalid username and password. 2. The application will prompt invalid credential. 3. Participant tries to login the the application with valid username and password. 4. Application login successfully.	Pass	No
UAT_2	Start Webcam	1. Participant clicks to access the webcam. 2. The application asks participants permission to access to the webcam.	Pass	No

		3. The webcam opens successfully.		
UAT_3	Stop Webcam	1. Participant clicks on the stop recording button. 2. Webcam close successfully	Pass	No
UAT_4	Show Prediction Output	1. Participant clicks to access the webcam. 2. The application asks participants permission to access to the webcam. 3. The webcam opens successfully. 4. Participant clicks on the start recording button. 5. The application shows the prediction output	Pass	No

Participant #	2			
Testing Date	24/04/2024			
UAT ID	Modules	Test Scenario	Results	Comments
UAT_1	Login	5. Participant tries to login the the application with invalid username and password. 6. The application will prompt invalid credential. 7. Participant tries to login the the application with valid username and password. 8. Application login successfully.	Pass	No
UAT_2	Start Webcam	4. Participant clicks to access the webcam. 5. The application asks participants permission to access to the webcam. 6. The webcam opens successfully.	Pass	No

UAT_3	Stop Webcam	3. Participant clicks on the stop recording button. 4. Webcam close successfully	Pass	No
UAT_4	Show Prediction Output	6. Participant clicks to access the webcam. 7. The application asks participants permission to access to the webcam. 8. The webcam opens successfully. 9. Participant clicks on the start recording button. 10. The application shows the prediction output	Pass	No

Participant #	3			
Testing Date	24/04/2024			
UAT ID	Modules	Test Scenario	Results	Comments
UAT_1	Login	9. Participant tries to login the the application with invalid username and password. 10. The application will prompt invalid credential. 11. Participant tries to login the the application with valid username and password. 12. Application login successfully.	Pass	No
UAT_2	Start Webcam	7. Participant clicks to access the webcam. 8. The application asks participants permission to access to the webcam. 9. The webcam opens successfully.	Pass	No

UAT_3	Stop Webcam	5. Participant clicks on the stop recording button. 6. Webcam close successfully	Pass	No
UAT_4	Show Prediction Output	11. Participant clicks to access the webcam. 12. The application asks participants permission to access to the webcam. 13. The webcam opens successfully. 14. Participant clicks on the start recording button. 15. The application shows the prediction output	Pass	No

Participant #	4			
Testing Date	25/04/2024			
UAT ID	Modules	Test Scenario	Results	Comments
UAT_1	Login	13. Participant tries to login the the application with invalid username and password. 14. The application will prompt invalid credential. 15. Participant tries to login the the application with valid username and password. 16. Application login successfully.	Pass	No
UAT_2	Start Webcam	10. Participant clicks to access the webcam. 11. The application asks participants permission to access to the webcam. 12. The webcam opens successfully.	Pass	No

UAT_3	Stop Webcam	7. Participant clicks on the stop recording button. 8. Webcam close successfully	Pass	No
UAT_4	Show Prediction Output	16. Participant clicks to access the webcam. 17. The application asks participants permission to access to the webcam. 18. The webcam opens successfully. 19. Participant clicks on the start recording button. 20. The application shows the prediction output	Pass	The prediction output is a bit slow.

Participant #	5			
Testing Date	25/05/2024			
UAT ID	Modules	Test Scenario	Results	Comments
UAT_1	Login	17. Participant tries to login the the application with invalid username and password. 18. The application will prompt invalid credential. 19. Participant tries to login the the application with valid username and password. 20. Application login successfully.	Pass	No
UAT_2	Start Webcam	13. Participant clicks to access the webcam. 14. The application asks participants permission to access to the webcam. 15. The webcam opens successfully.	Pass	No

UAT_3	Stop Webcam	9. Participant clicks on the stop recording button. 10. Webcam close successfully	Pass	No
UAT_4	Show Prediction Output	21. Participant clicks to access the webcam. 22. The application asks participants permission to access to the webcam. 23. The webcam opens successfully. 24. Participant clicks on the start recording button. 25. The application shows the prediction output	Pass	No

Appendix D: Kanban Board