Object Detection for

Inventory Stock Counting

By

Babila, Isaiah Francis E.

Villasor, Shawn Antonie E.

Electronics and Communications Engineering 3rd Year

A Research Proposal Submitted to the School of EECE in Partial Fulfilment of the Requirements for the Degree

Bachelor of Science in Electronics Engineering

Mapua University

October 2021

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: INTRODUCTION

Stock control or inventory management is essential in business, keeping the stocks accounted for depends on knowing the inventory, and knowing where all the supplies are located [1], instead of hiring employees to spend time sifting through files, sending spreadsheets to one another, manually writing all reports, or visiting the storage room when there is uncertainty regarding stock. The study of object counters avoids these altogether using a sound inventory management system. Workers will be more productive if they can spend more time on more valuable tasks that contribute to their company's or business' bottom line, and it will be more efficient with more accessibility to inventory information. The manual method of inventory counting can be erroneous like any other manual process, can be time consuming, unproductive, and can be resource taxing. Because the manual method may cause problems, a study on stock counting applying image processing techniques was conducted to avoid errors in maintaining inventory of products [2]. Developing object detection software has never been easier. These techniques have also used massive image algorithms to reduce the need for large datasets, which has resulted in significant performance improvements. Furthermore, performance has improved significantly because current techniques rely on full womb-to-tomb processes, allowing for real-time use cases. [3]. You Only Look Once (YOLO) is an object detection algorithm that uses a single convolutional network. Unlike all other object detection algorithms that sweep the image bit by bit, the algorithm takes the entire image. It reframes it as a single regression challenge, heading directly from image pixels to bounding box coordinates and class probabilities [4]. With the rapid evolution of image recognition technology in recent years, the software now implements image recognition techniques to calculate the object through the camera. After the algorithm has identified all of the items in the video, the system should determine the relevance of the product detected in various frames to achieve inventory stock counting [5].

According to an article, there are challenges in detecting objects that considerably reduce object detection performance. These challenges are the variable number of objects in an image, the main issue with this is in Machine Learning models. We wouldn't know the correct number of outputs because the number detected in the Image is unknown beforehand. Another is multiple spatial scales and aspect ratios. It isn't easy to track down objects because of their multiple dimensions. Another challenge is modeling; the loss function's dual behavior frequently performs poorly in both. A limited amount of annotated data is another problem for object detection. Gathering the ground truth labels and the bounding box for each class remains a time-consuming task. And Speed for real-time detection is typically a video shot at almost 24 frames per second (fps) and developing an algorithm that can achieve that frame rate is difficult [6]. Some algorithms have attempted to solve these problems before revealing the best of YOLO Algorithms. These are the Fast R-CNN, Single Shot MultiBox Detector (SSD), and the Retina-Net. These algorithms have solved some of the challenges mentioned, except the most important one, the Speed for real-time object detection.

The YOLO algorithm provides better performance on all mentioned parameters and a high fps for real-time use. Instead of selecting the most exciting image, the YOLO algorithm predicts classes and bounding boxes for the entire image in a single algorithm run. To comprehend the YOLO algorithm, we must first understand what is being projected. Eventually, we want to accurately predict an object and the bounding box that identifies its location [6]. The need for the study is real-time counting, multiple object sensing, and object classification. An object counting system needs to be done in real-time. Also, it needs to be capable of detecting multiple objects to count the products accurately. Lastly, the object counting system must distinguish objects from other items because there will be noise that the camera might sense. One of the major problems encountered in the cellphone business is the counting of stocks. This is normally done by a person and takes up a lot of time and resources. This study would implement an automatic inventory stock counting which would be done by a camera seeing the stocks with object detection algorithms.

The main objective of this study is to apply object detection algorithm to identify and count Cherry mobile phones Aqua S9 and Cherry Flare S8 models placed in a storage cabinet. Specifically, this study will: (1) Build a database at different orientations such as (a) normal orientation (0° from the z-axis), (b) Upside-Down (180°), and lastly (c) Sideview (90° and -90° from the x-axis). (2) Identify differences between a 5W, 7W, and 12W light source effect to detection. (3) Apply YOLOv5S to identify and count the items and display the result in the touch display (4) Validate the system's accuracy in a makeshift shelves setup.

This study only focuses on counting and detecting cell phone boxes using digital image processing techniques and learning algorithms using the YOLO. It is to calculate the number of products and automatically help business owners in their stocks inventory. This will let them know the number of their stocks using the camera installed in their storage or cabinet. As a standard to inventory management, stock or inventory counting should be done initially on a weekly basis until a stable inventory count is achieved and later reduced to a monthly frequency. As the process matures and the compliance of the stakeholders executing the inventory process becomes part of their normal routine, then a review of the frequency of performing inventory counting can be done to assess if it can be reduced further to quarterly.

This study will cover recommendations for inventory management for Cherry Mobile models, namely, Aqua S9 and Cherry Flare S8. The recommendations of this research shall also include how the other noises due to object orientation and lighting. This study will only use YOLO as the object detection algorithm to be used by the camera. This research shall not cover any other proposal on how the YOLO can help in monitoring other phone models, other items such as grocery products or any other merchandise items. The packaging model that would be used in this study would be the packaging for the Aqua S9 2021 model and the Cherry Flare S8 2021 model. Changing packages would be for another study or another machine learning study.

CHAPTER 2: REVIEW OF RELATED LITERATURE AND STUDIES

This chapter contains relevant literature and studies that the researcher considered to substantiate the importance of this research study. It also includes a state-of-the-art to understand the research for better comprehension of the study entirely.

2.1 Inventory Stock Counting

A sound inventory control system is needed to reduce costs and stay competitive [7]. Admittedly, one of the inventory control strategies is stocktaking. This implies that the control application is also developed to keep inventory, record transactions, generate cycle counting schedules and perform the cycle counting. It shows by using the stocktaking policy of cycle counting, which is poured into an inventory control program, with the expectation of increased inventory record accuracy in the pharmaceutical industry.

2.1.1 Supply Chain Management

Maintaining an accurate and close to the real-time inventory of items is crucial for effective Supply Chain Management (SCM), one of the essential cornerstones of successful retail company decisions [8]. Interestingly, perpetual inventory systems are not enough to accurately picture the current inventory due to theft and misplacement. This implies that manual stocks using handheld RFID readers tend to be tedious, expensive, and inaccurate even if the merchant has installed an RFID-based solution. This shows that the solution can autonomously take stocks with high accuracy is expected to significantly impact the market.

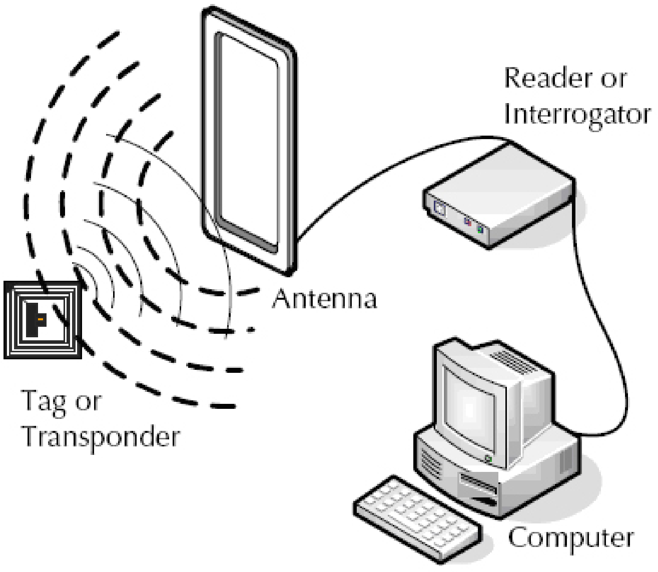


Figure 2.1 RFID System Works

2.1.2 Supply Chain Network

An industrial machine vision system is introduced for effective inventory maintenance to minimize the production cost in the supply chain network (SCN). Consequently, the appearance model is considered as a visual signature by which individual objects can be detected anywhere via a camera feed. This explains why Speeded Up Robust Features (SURF) features are extracted from prototype image, which refers to a predefined template of particular things, and then extracted from the camera feed of inventory [9].

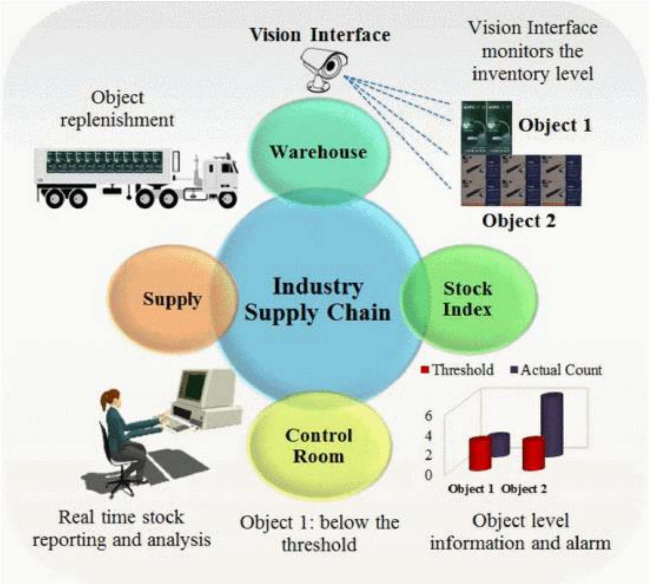


Figure 2.2 Supply Chain Network for Inventory Control

Nowadays, warehousing and storage subsector industries require an efficient management system to perform accurate inventories, optimize operations, and reduce costs. This explains why that routinely affects warehouse operations, includes keeping track of stock in real-time by counting how many items leave or enter the warehouse and those currently stored in the warehouse. At least two employees are required to verify and count the stock in real-time and then confirm the accuracy of the data. This is not only a time-consuming process for employees, but it is also susceptible to human error, causing the loss of revenue, and ultimately, a negative outcome for the company [10].

2.2 Object Detection Convolutional Neural Network (CNN)

With the development of intelligent devices and social media, the data bulk on Internet has grown with high speed [11]. Admittedly, it is an essential aspect of image processing, and object detection has become one of the famous international research fields. This shows the powerful ability with feature learning and transfer learning of Convolutional Neural Network (CNN) has received growing interest within the computer vision community, thus making a series of essential breakthroughs in object detection. So, it is a significant survey on how to apply CNN to object detection for better performance.

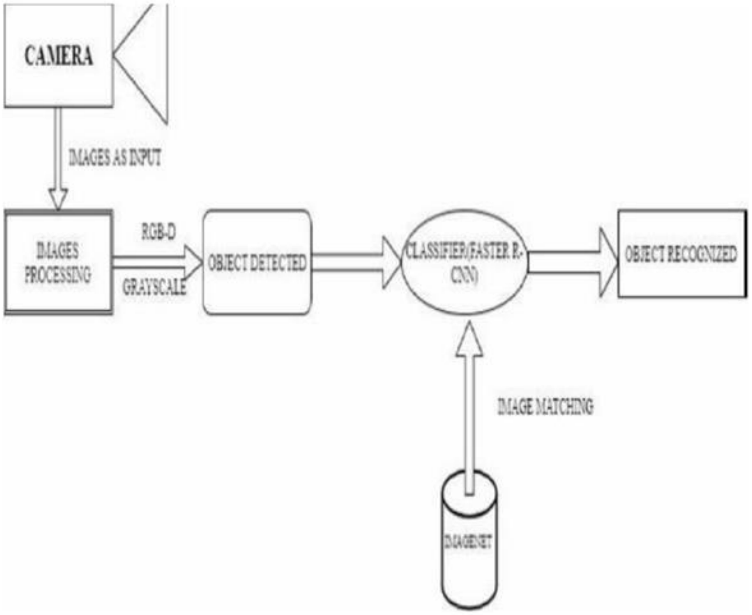


Figure 2.3 CNN System Flowchart

2.2.1 Deep Learning Architecture

Deep learning has evolved as a powerful machine learning technique that incorporates multiple layers of features or data representations and provides state-of-the-art results. Interestingly, the application of deep learning has shown impressive performance in various application areas, particularly in image classification, segmentation, and object detection. This explains the focus on the application of deep learning architectures to three major applications, namely:

- Wild animal detection,
- Small arms detection

- Human being detection.

It means that the detailed review summary, including the systems, database, application, and accuracy claimed, is also provided for each model to serve as guidelines for future work in the above application areas [12].

2.2.2 Object Detection

With the recent advancement in deep neural networks in image processing, classifying and detecting the object accurately is now possible. This implies that the Convolutional Neural Networks (CNN) are used to detect objects in the environment. Two state-of-the-art models are compared for object detection, Single Shot Multi-Box Detector (SSD) with MobileNetV1 and a Faster Region-based Convolutional Neural Network (Faster-RCNN) with InceptionV2. It shows that one model is ideal for real-time application because of speed, and the other can be used for more accurate object detection [13].

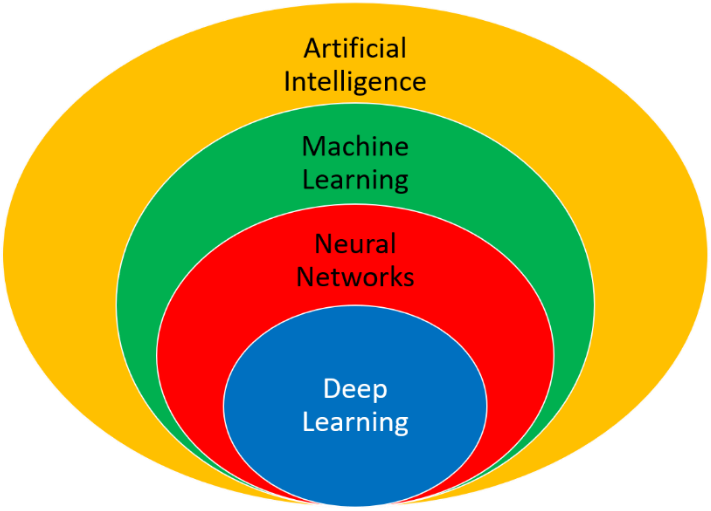


Figure 2.4 Deep Learning

The state-of-the-art performance for object detection has been significantly improved over the past two years. Interestingly, the introduction of powerful deep neural networks, such as Google Net and VGG, novel object detection frameworks, such as R-CNN and its successors, Fast R-CNN and Faster R-CNN, play an essential role in improving state-of-the-art [14]. Despite their effectiveness on still images, those frameworks are not explicitly designed for object detection from videos.

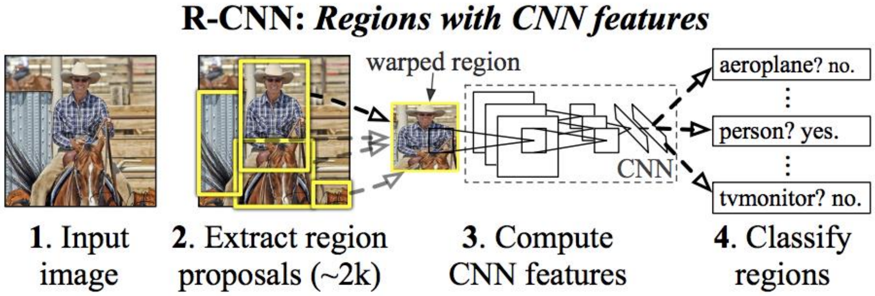


Figure 2.5 R-CNN Features

Object detection is a primary issue of very high-resolution remote sensing images (RSIs) for automatically labeling objects [15]. This explains why deep learning has gradually gained the competitive advantage for remote sensing object detection, mainly based on convolutional neural networks (CNNs). Local knowledge can provide spatial data, which is helpful for accurate localization. In addition, there are variable factors, such as rotation and scaling, which affect the object detection accuracy in RSIs.

2.3 You Only Look Once (YOLO)

YOLO is a practical, one-stage tool for object detection and classification. In this paper, we consider the application of grocery product detection [16]. Interestingly, grocery stores have many product classes, so it is beneficial to postpone the classification into a second, specialized neural network with a higher capacity. It appears that extracting bounding boxes for a classification network is not precise enough as the detected area includes redundant information about the background. YOLO reaches high detection precision by using YOLO a prior knowledge, anchors extracted from data.

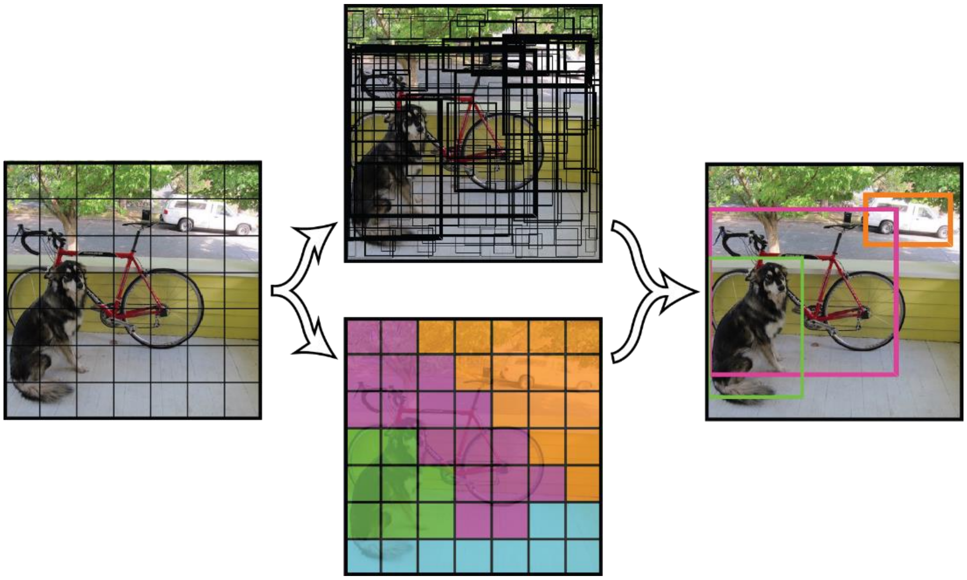


Figure 2.6 Bounding Box in YOLO Model

A deep learning algorithm based on the you-only-look-once (YOLO) method is developed (PCBs). Interestingly, deep learning algorithms' high accuracy and efficiency have increased their adoption in every field. Admittedly, accurate detection of PCB defects using deep learning algorithms, such as convolutional neural networks (CNNs), has garnered considerable attention. This shows that the highly skilled quality inspection engineers first use an interface to record and label defective PCBs [17].

Object detection, which combines object categorization and object location within a scene, is regarded as one of the most difficult challenges in this subject of computer vision. Consequently, deep neural networks (DNNs) have been shown to outperform previous approaches in terms of object detection, with YOLOv2 (an upgraded You Only Look Once model) being one of the most advanced DNN-based object detection methods in terms of speed and accuracy [18]. Although YOLOv2 can achieve real-time performance on a powerful GPU, it remains challenging to leverage this approach for real-time object detection in the video on embedded computing devices with limited computational power and memory. We present a new framework in this research called Fast YOLO, a fast You Only Look Once framework which accelerates YOLOv2 to be able to perform object detection in the video on embedded devices in a real-time manner. It shows that the proposed Fast YOLO framework can reduce the number of deep inferences by an average of 38.13% and an average speed up of ~3.3X for objection detection in the video compared to the original YOLOv2, leading Fast YOLO to run an average of ~18FPS on an Nvidia Jetson TX1 embedded system.

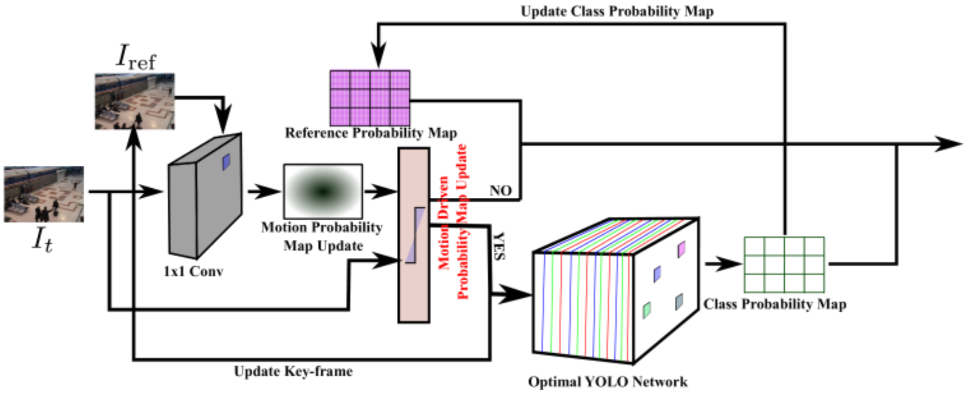

Update Class Probability Map
I_{ref}
Reference Probability Map
Motion Driven Probability Map Update
NO
YES
I_t
1x1 Conv
Motion Probability Map Update
Class Probability Map
Update Key-frame
Optimal YOLO Network

Figure 2.7 Proposed Fast YOLO Framework for Object Detection in Video

YOLO presented as a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. It frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. This may be because a single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. It can be deduced that the YOLO model processes images in real-time at 45 frames per second. Fast YOLO's more petite version of the network processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. It outperforms other detection methods, including DPM and R-CNN, when generalizing raw images to other domains like artwork [19].

Using a boosted and modified version of the You Only Look Once (YOLO) concept, a high-performance algorithm was developed based on the TensorFlow framework, enhances the real-time monitoring of traffic-flow problems by an intelligent transportation system [20]. Using a boosted and modified version of the You Only Look Once (YOLO) concept, a high-performance algorithm was developed and used for training. It reveals that the YOLO-accuracy UA's was 22 percent higher than the YOLO model before optimization for various weather circumstances, and the recall rate increased by roughly 21 percent. The YOLO-UA model's accuracy, precision, and recall rate were more than 94 percent higher than the floating rate, indicating that precision and recall rate had reached a good equilibrium. When employed for video testing, the YOLO-UA model produced traffic data with a precision of up to 100% and a count time of fewer than 30 milliseconds.

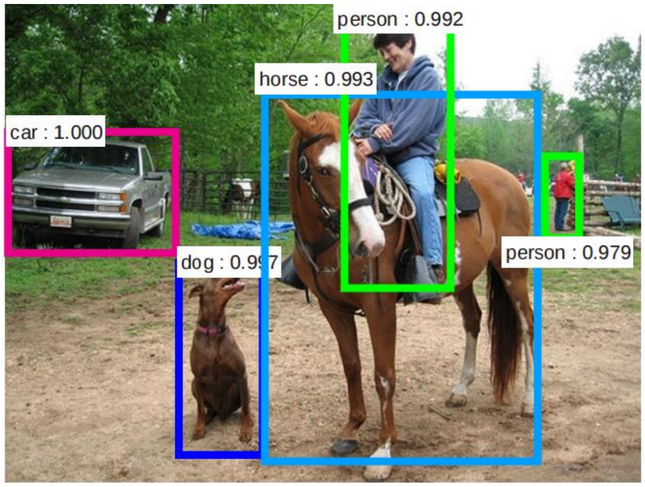


Figure 2.8 Object Detection with TensorFlow API

2.4 Single Shot Detector (SSD)

Single Shot Detector is a state-of-the-art object detection algorithm that combines high detection accuracy and real-time performance. [21]. It is widely recognized that SSD is less accurate in detecting small objects than large objects because it ignores the context outside the proposal boxes.

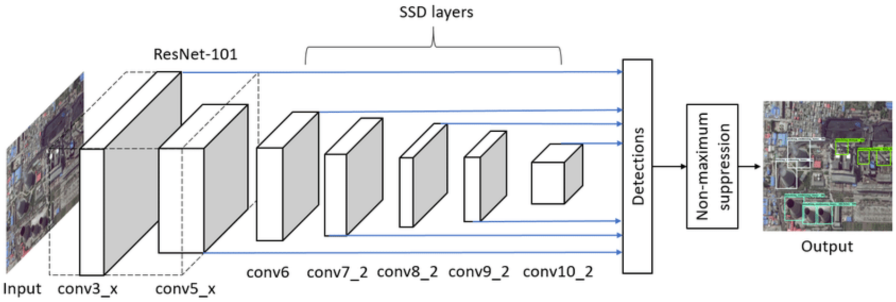


Figure 2.9 Framework of Single Shot Detector

Traffic density estimation is an essential component of an automated traffic monitoring system. SSD can handle different shapes, sizes, and view angles of objects. It shows that Mobile Net-SSD is a cross-trained model from SSD to Mobile Net architecture, faster than SSD. In this study, it can be concluded that the SSD framework shows significant potential in the field of traffic density estimation. SSD achieved 92.97% average detection accuracy in our experiment. On the other hand, the Mobile Net-SSD achieved 79.30% average detection accuracy [22].

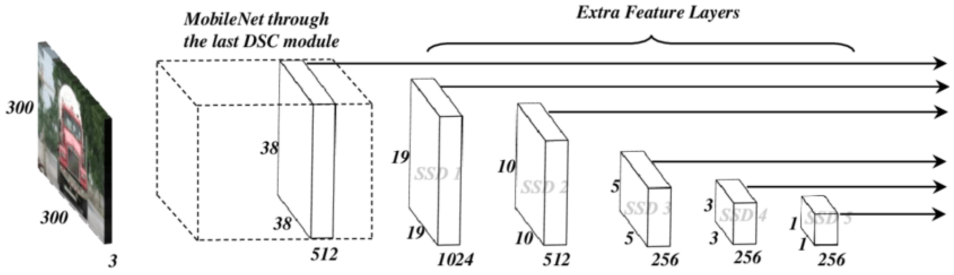


Figure 2.10 Mobile Net-SSD Network Architecture

One of the most accurate and fast object detection algorithms available is Single Shot Multibox Detector. [23]. SSD's feature pyramid detection method makes it hard to fuse the features from different scales. This explains why, in the feature fusion module, elements from various layers with different scales are concatenated together, followed by some down-sampling blocks to generate a new feature pyramid, which Multibox detectors will feed to predict the final detection results.

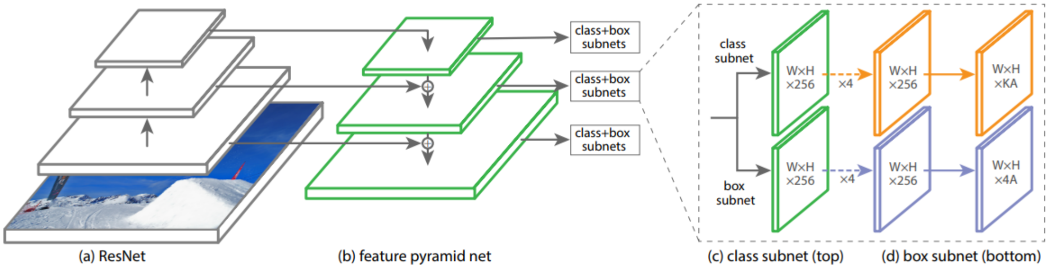


Figure 2.11 Feature Pyramid Network

Vehicle detection plays an influential and essential role in traffic safety, attracting extensive attention from academics and the industry. Admittedly, deep learning has made significant breakthroughs in vehicle detection applications. It shows that the Single Shot Detector (SSD) algorithm, one of the object detection algorithms, is used to detect vehicles. It is expected that its main challenge is that the computing complexity and low accuracy [24].

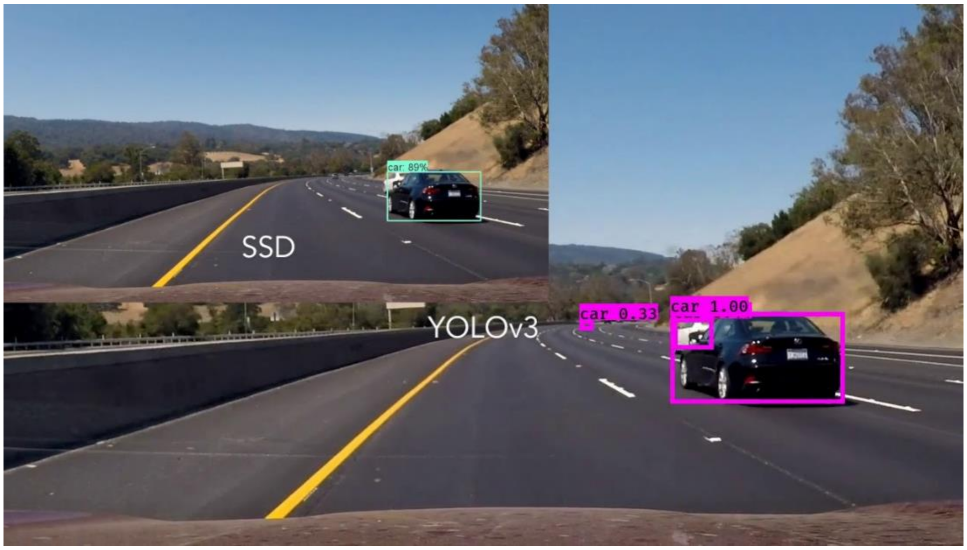


Figure 2.12 Vehicle Detection with YOLOv3 and SSD

The Single Shot Multi-Box Detector (SSD) detects an object in an image using a single convolutional neural network, is one of the fastest algorithms in the current object detection field. As a result, SSDs are quick. When compared to the state-of-the-art on mAP, there is a significant gap [25]. Based on the Inception block to replace the extra layers in SSD, call this method Inception SSD (I-SSD). It shows the proposed network can catch more information without increasing the complexity. This implies the batch-normalization (BN) and the residual structure in our I-SSD network architecture. It means that the I-SSD algorithm achieves 78.6% mAP on the Pascal VOC2007 test, which outperforms the SSD algorithm while maintaining its time performance.

Detecting objects is a difficult task in computer vision, involving both object classification and object localization within a scene-based form the stud of [26]. Deep neural networks have been shown to yield effective techniques for overcoming the obstacle of object detection. Small deep neural network architectures for object detection that are more suitable for embedded devices, such as Tiny YOLO and SqueezeDet, are becoming increasingly popular. The resulting Tiny SSD possesses a model size of 2.3MB (~26X smaller than Tiny YOLO) while still achieving a mAP of 61.3% on VOC 2007 (~4.2% higher than Tiny YOLO). These experimental results show that real-time object detection well-suited for embedded scenarios can design minimal deep neural network architectures.

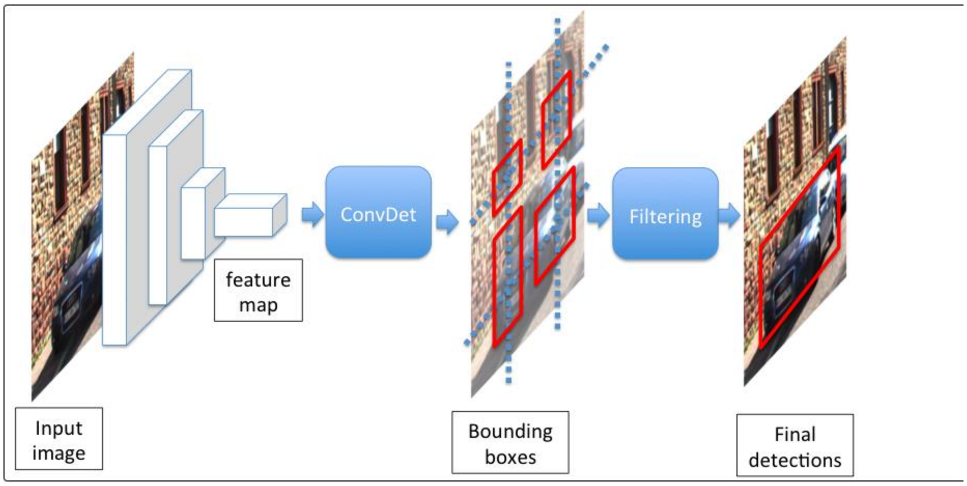


Figure 2.13 SqueezeDet Model

2.5 Spatial Pyramid Pooling (SSP-net)

The Scalable Sequential Pyramid Networks (SSP-Net) as it is trained with refined supervision at multiple scales in a sequential manner [27]. It demonstrates that the network only needs one training procedure and can make the best predictions at 120 fps or acceptable forecasts at more than 200 FPS when tested. The method can perform multi-resolution intermediate supervisions and achieve results comparable to the state-of-the-art with very low-resolution feature maps because the proposed regression approach is invariant to feature map size. Extensive testing on Human3.6M and MPI-INF-3DHP, two of the most important publicly available datasets for 3D pose estimation, was used to determine our method's accuracy and effectiveness. This shows the relevant insights about our decisions on the network architecture and its flexibility to meet the best precision-speed compromise.

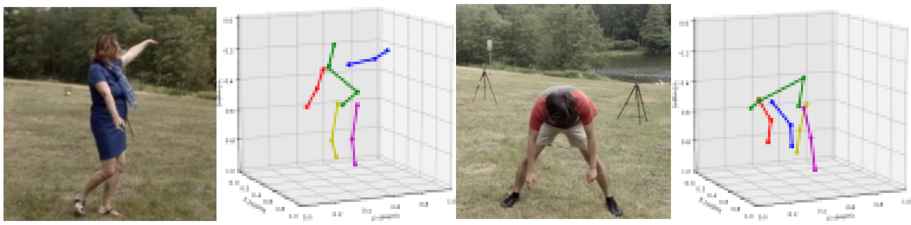


Figure 2.14 3D poses (bottom) for the MPI-INF-3DHP dataset

Ship detection has played a significant role in remote sensing for a long time, but it is still full of challenges. This is attributed to the fact that Traditional ship detection methods are typically limited by the difficulty of intensive object detection, the complexity of application scenarios, and the detection region's redundancy proposed a framework Rotation Dense Feature Pyramid Networks (R-DFPN) that can effectively detect ships in various scenes, including the ocean and port [28]. This implies the case of ship rotation and lush arrangement; it designed a rotation anchor strategy for predicting the object's minimum circumscribed rectangle to reduce the redundant detection region and improve the recall.

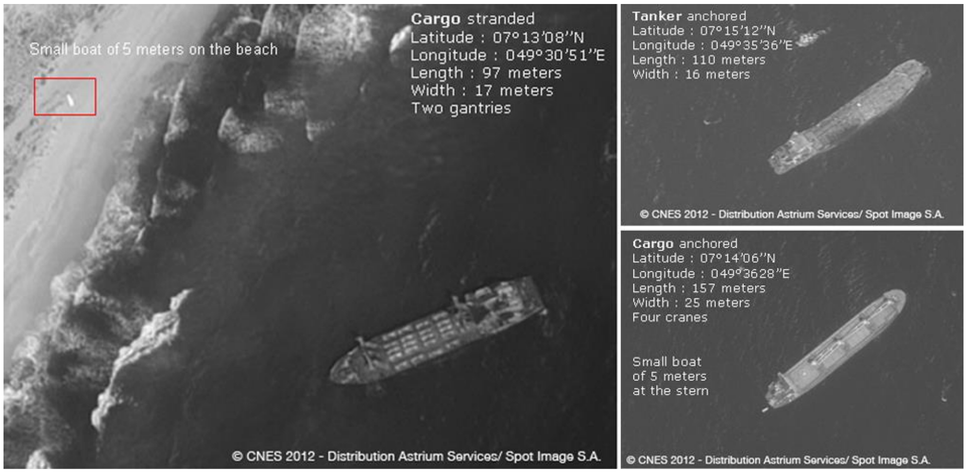


Figure 2.15 Ship detection from remote sensing

Vehicle detection is an essential component in unmanned driving systems. The real-time vehicle detection method under complex road conditions solves the real-time detection of road vehicles in automatic driving.

1.) Using the YOLO network to build a deeper depth neural network, which is used to identify and score the vehicles in the image.

2.) The dynamic threshold method will use the dynamic threshold method to remove some false candidate boxes and the Gauss attenuation function to filter the overlapping candidate boxes.

3.) The loss function normalized by the change rate is used to train the neural network.

4.) The batch normalization layer is used to correct the input data of the network to avoid data deviation during the training process.

5.) An entire convolution layer is added at the end of the network layer to transform the two-dimensional data into one-dimensional data and output the final recognition results.

It is verified that this method improves the efficiency and accuracy of real-time vehicle detection. It can effectively detect vehicles on roads with complex backgrounds and satisfy the real-time requirements of road vehicle detection [29].

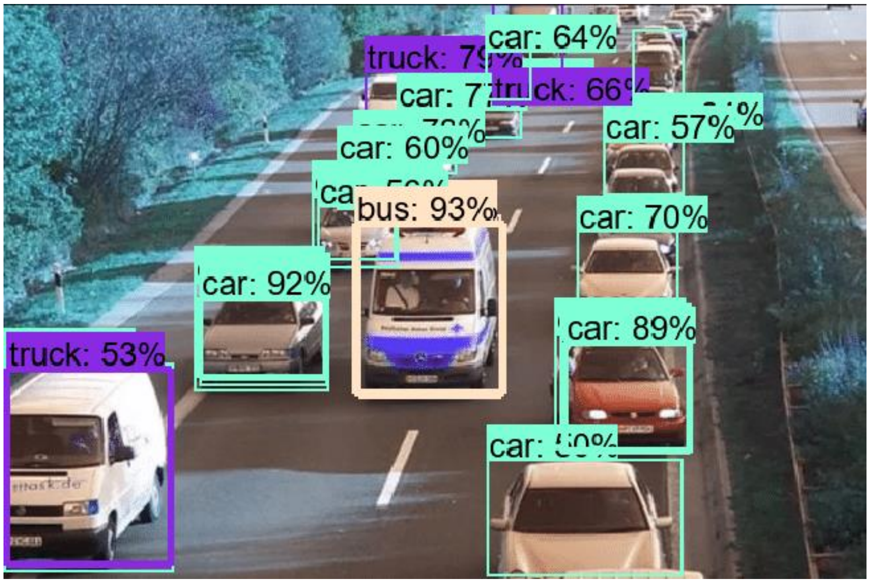


Figure 2.16 Example of Vehicle Detection by YOLO Object Detector

Bounding box algorithms are helpful in the localization of image patterns. Interestingly, the utilization of convolutional neural networks on X-ray images has proven a promising disease prediction technique. It shows pattern localization over prediction has always been a challenging task with inconsistent coordinates, sizes, resolution, and capture positions of an image. Several model architectures are used for object detection. Localization in modern-day computer vision applications like Fast R-CNN, Faster R-CNN, Histogram of Oriented Gradients (HOG), You only look once (YOLO), Region-based Convolutional Neural Networks (R-CNN), Region-based Fully Convolutional Networks (R-FCN), Single Shot Detector (SSD), etc. SSD and region-based detectors like Fast R-CNN or Faster R-CNN are very similar in design and implementation. Still, SSD has been shown to work efficiently with larger frames per second (FPS) and lower resolution images [30].

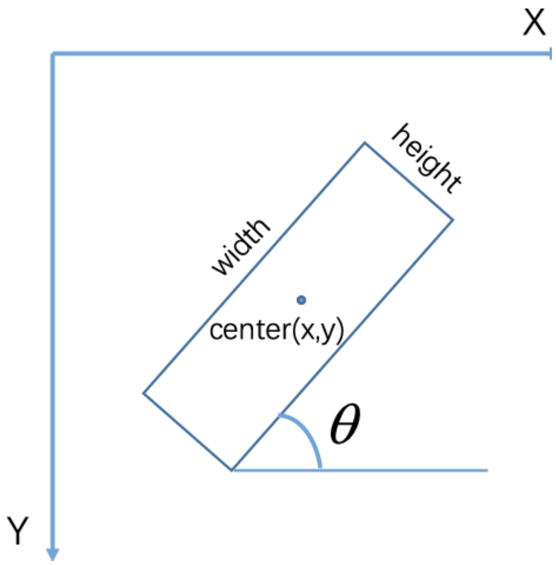


Figure 2.17 General Representation of Rotation Bounding Box

Synthesis of the study

The researched literature and studies, associated directly and indirectly with the present research, provided more insights about the nature and scope of the study, which is the feasibility of the different Cherry Mobile stores in Metro Manila.

Both literature and studies discussed object detection and automatic inventory stock counting for cellphone selling businesses that used you only look once (YOLO) algorithm. The study of [7] stressed that a good inventory control system needs the organization to reduce costs and stay competitive. Admittedly, one of the inventory control strategies is stock-taking. This implies that the control application is also developed to keep inventory, record transactions, generate cycle counting schedules and perform the cycle counting. It shows that using a stock-taking policy of cycle counting, which is poured into the inventory control application, may improve the inventory record accuracy in the pharmaceutical company.

On the other hand, the Object Detection Convolutional Neural Network (CNN), according to [11], stressed the development of intelligent devices and social media. The data bulk on the Internet has grown with high speed. Admittedly, it is an essential aspect of image processing, and object detection has become one of the famous international research fields. This shows the powerful ability with feature learning and transfer learning of Convolutional Neural Network (CNN) has received growing interest within the computer vision community, thus making a series of essential breakthroughs in object detection. So, it is a significant survey on how to apply CNN to object detection for better performance.

However, a deep learning algorithm based on the you-only-look-once (YOLO) approach is proposed for the quality inspection of printed circuit boards (PCBs). Interestingly, deep learning algorithms' high accuracy and efficiency have resulted in their increased adoption in every field. Admittedly, accurate detection of PCB defects using deep learning algorithms, such as convolutional neural networks (CNNs), has garnered considerable attention. This shows that the highly skilled quality inspection engineers first use an interface to record and label defective PCBs. It shows that they achieved a defect detection accuracy of 98.79% in PCBs with a batch size of 32 [17].

These related literature and studies have a significant relationship with the present study. Both discuss the benefits of object detection and automatic inventory stock counting for cellphone selling businesses using you only look once (YOLO) algorithm. That could enhance the cellphone business of sale used by the cellphone stores. Though at some point, selling cellphones and other gadget businesses to avoid illegal acts must consider some rules and regulations.

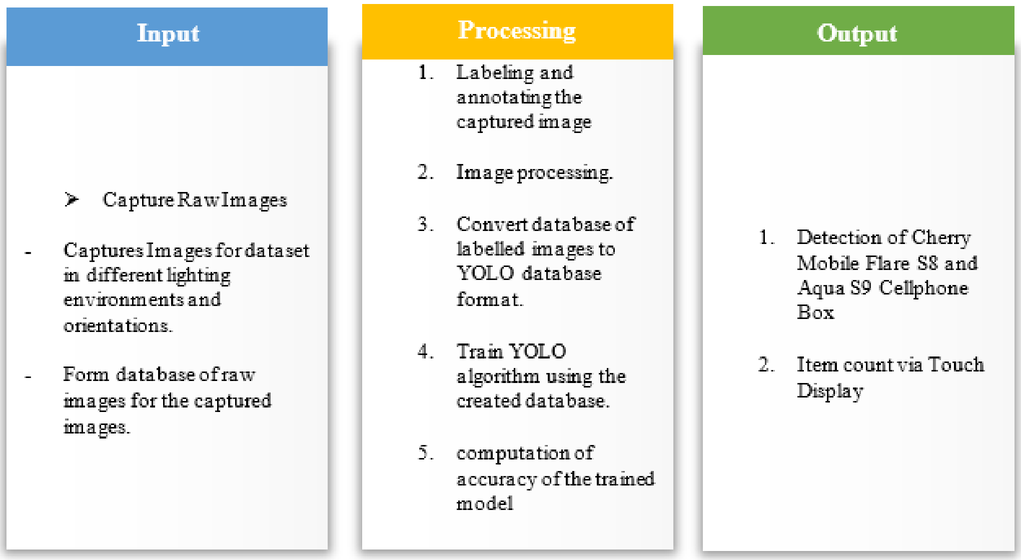
Chapter 3: METHODOLOGY

This chapter concentrates on the discussion of the research methods and procedures used by the researchers to answer the general questions imposed by the study and obtain the objectives of the study. Specifically, the research method, research design, source of data, data gathering procedure, data analysis and research instrument that will be used for accurate data analysis and interpretation were explained in this chapter.

3.1. Research Method and Techniques Used

The thesis undergoes an experimental method of research. This type of study would gather data from the images and videos captured by the camera that would be used in this study and would also further be discussed in this thesis. Data gathered primarily acquired by the researchers, through systematic approach manipulating variables under set conditions. With that the statistical confidence level is set to be 70% and above to attain the accuracy within circumstances.

3.1.1. Conceptual Framework



Input

➢ Capture Raw Images

- Captures Images for dataset in different lighting environments and orientations.
- Form database of raw images for the captured images.

Processing

1. Labeling and annotating the captured image
2. Image processing.
3. Convert database of labelled images to YOLO database format.
4. Train YOLO algorithm using the created database.
5. computation of accuracy of the trained model

Output

1. Detection of Cherry Mobile Flare S8 and Aqua S9 Cellphone Box
2. Item count via Touch Display

Figure 3.1 System Framework

Figure 3.1 shows the conceptual framework of the proposed topic and is divided into three sections. These are input, processing, and output. The input section will collect data from the database to vary the image parameters, capture raw images for a dataset in different lighting environments and orientations. For the processing part, this is to label the target objects in the image in a bounding box, resizing the raw image to squares aspect ratio convert database to YOLO database format, train the YOLO algorithm using the made database, and compute the accuracy of the trained model. Lastly, the video captured from the Raspberry Pi hardware will recognize the trained model and detect and count the target objects that will be displayed to the touch display.

3.1.2. You Only Look Once

Joseph Redmon released the first model in 2016, followed by YOLOv2 (2017) and YOLOv3 (2018). In 2020, Joseph Redmon left the project due to ethical concerns in the computer vision field, and his work was improved by Alexey Bochkovskiy, who released YOLOv4 in the same year [31].

The algorithm is a real-time object detection system that is state-of-the-art. It processes images at 30 frames per second on a Pascal Titan X and has a mean Average Precision (mAP) of 57.9% on COCO test-dev [32].

- This algorithm applies the model to an image at different scales and locations. The high-scoring regions of the image are referred to as detections.
- This algorithm takes an entirely new approach. It uses a single neural network to process the entire image. This network divides the image into regions, which predicts bounding boxes and probabilities for each area. The predicted probabilities are used to weigh these bounding boxes.

- In comparison to classifier-based systems, YOLO has several advantages. It examines the entire image at test time, so the image's overall context impacts its predictions. It can also make predictions based on single network analysis. This makes it much faster than other object detection algorithms, more than 1000 times faster.

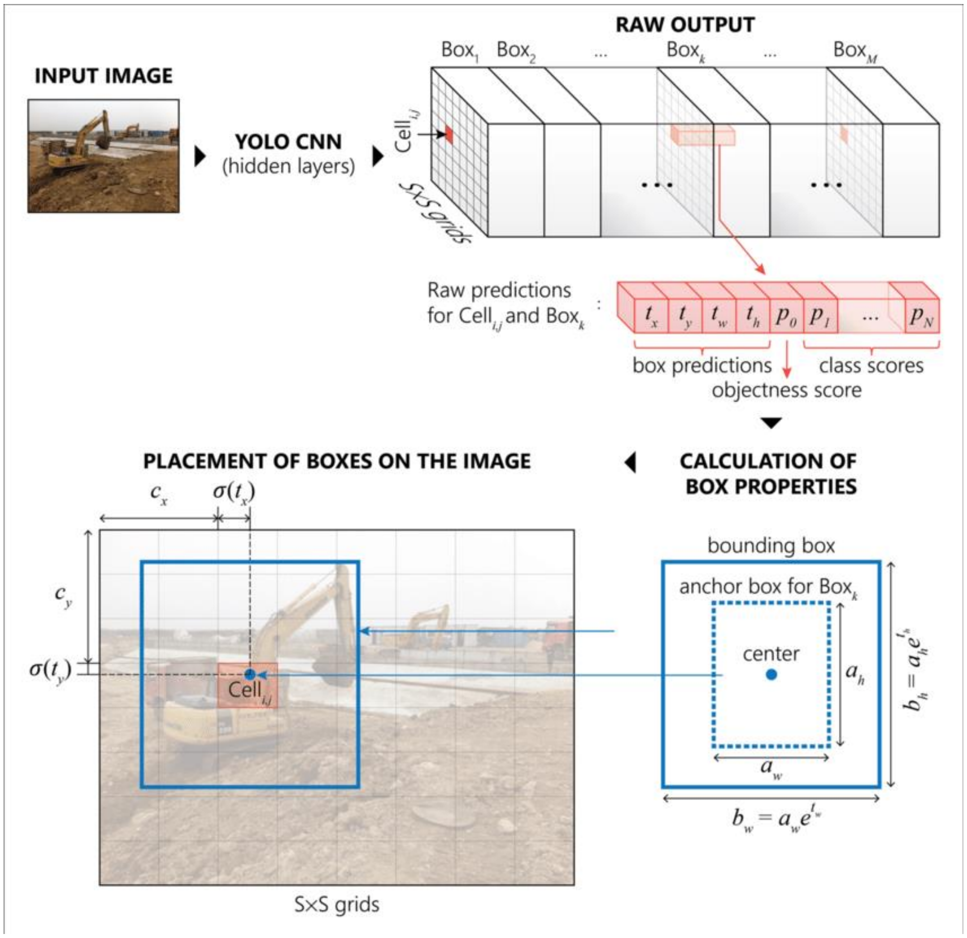


Figure 3.2 Schematic Diagram of the YOLO algorithm [33].

3.1.3 YOLOv5 Object Detection

YOLOv5 is the next contentious member of the YOLO family, released by Ultranytics in 2020, just days after YOLOv4.

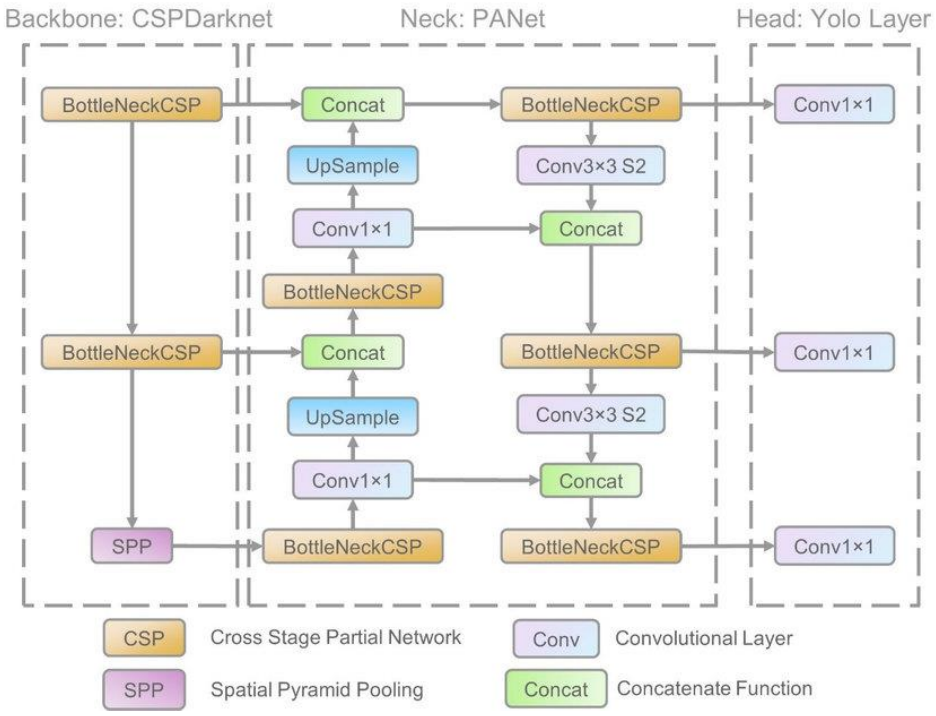


Figure 3.3 YOLOv5 Architecture

Backbone, Neck, and Head are the three main architectural blocks model of the YOLO family.

- YOLOv5 Backbone: It is used to extract features from images made up of cross-stage partial networks.
- YOLOv5 Neck: It generates a feature pyramids network using PANet to perform feature aggregation and passes it to Head for prediction.
- YOLOv5 Head: Layers that generate object detection predictions from anchor boxes.

The difference between YOLOv5 to the other versions of YOLO is directory structure and storage size. Previous versions of Yolo required the path to two different directories containing the images and their annotations when using custom data. For the old version stores, the weights least are 250mbs. At the same time, the YOLOv5 S version has a 27MB weight file [31].

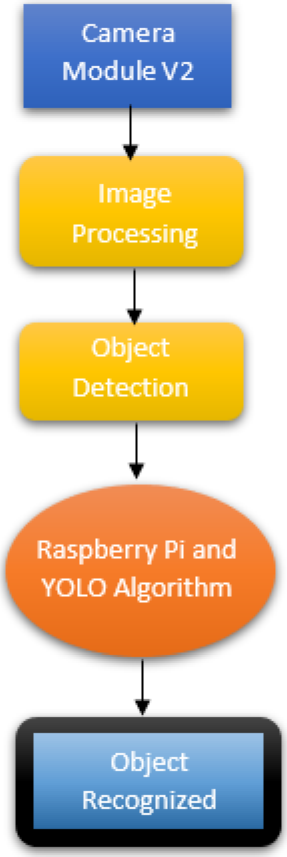


Figure 3.4 Camera Module V2 and Raspberry Pi Proposed System for Object Detection.

Refer to figure 3.4 for an illustration of the proposed system for object detection and counter. For detection, the camera module captures an image of the target object. After that, image processing takes the image as raw input data, processes it through object detection, and then uses a specified algorithm to transform it into more helpful information that humans can comprehend. Then, the image is detected and counted by its application.

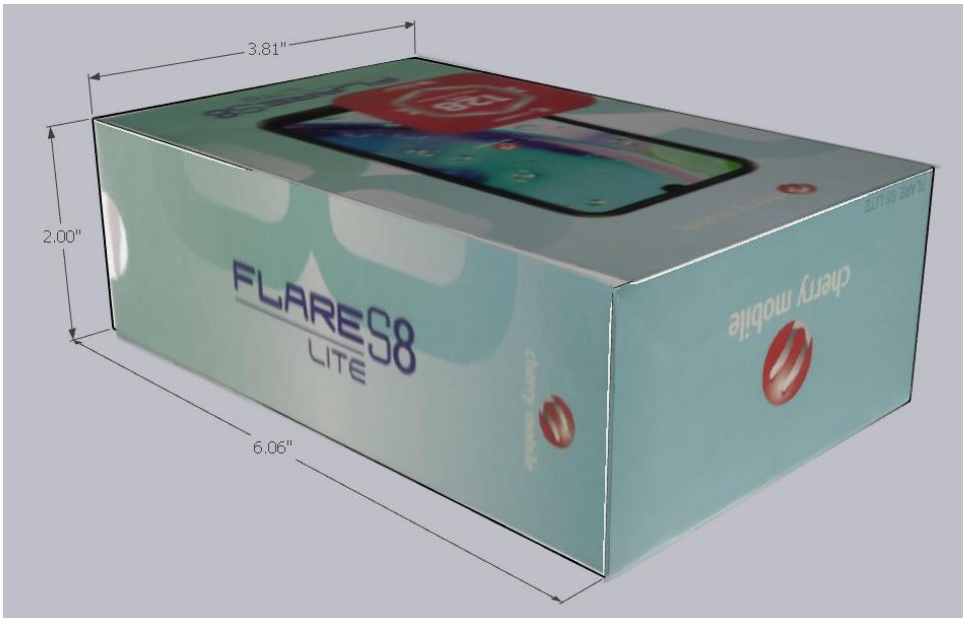


Figure 3.5 Object 1 Cherry Mobile Flare S8 Lite Dimension

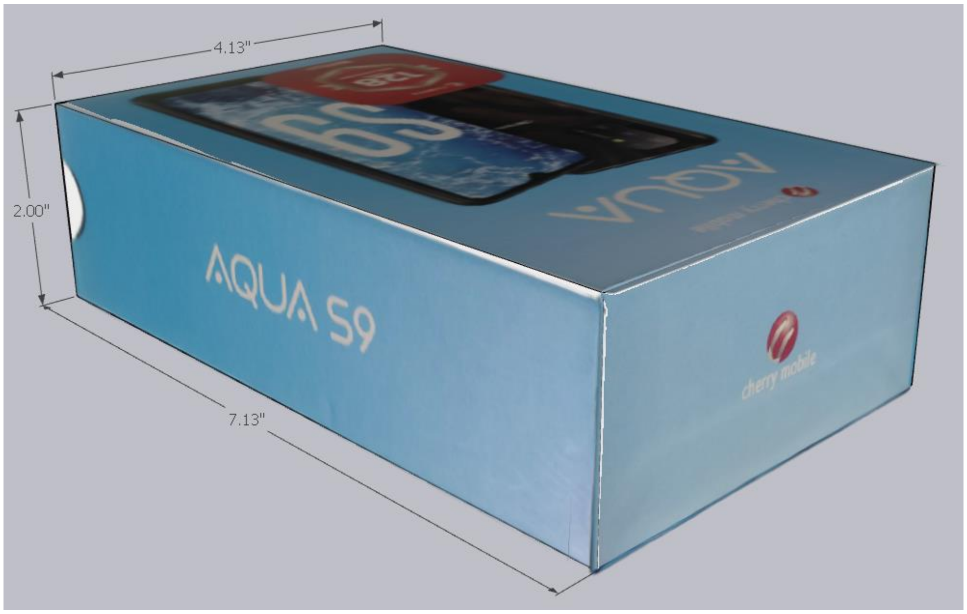
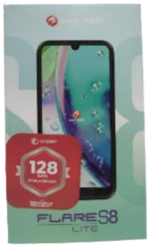
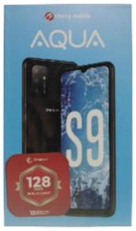


Figure 3.6 Object 2 Cherry Mobile Aqua S9 Dimension

Table 3.1 Object Characteristic

Object	Name	Dimension	Area	Shape	Color	Product Description
	Cherry Mobile Flare S8 Lite	6.0625x3.8125x2in	46.23 in^3	Cuboid	Label name: Mulled wine Brand name: White Body: Bay Leaf, Opal, White	Cellphone box
	Cherry Mobile Aqua S9	7.125x4.125x2in	58.78in^3	Cuboid	Label name: White Brand name: White Body: Hippie blue	Cellphone box

A dataset of images pertaining to the object to be classified in the algorithm is required, as shown in Table 3.1. In this paper, we'll go over how to prepare the images for data training as well as how to train the algorithm itself. The subject for object detection is specified in the table. Each object's characteristics are broken down into name, ratio, area, shape, color, and product description. This will provide crucial information for the algorithm's foundation.

3.2 Data Gathering

3.2.1 Data Gathering Flowchart

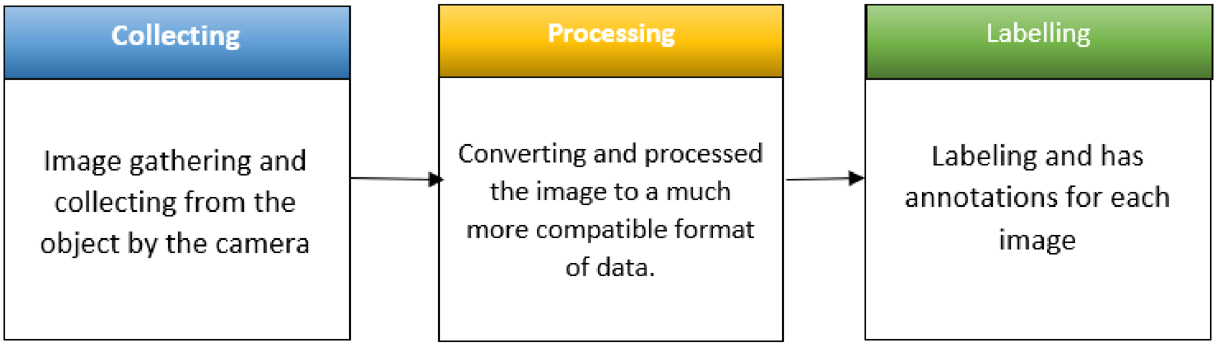

Collecting
Image gathering and collecting from the object by the camera
Processing
Converting and processed the image to a much more compatible format of data.
Labelling
Labeling and has annotations for each image

Figure 3.7 Image Gathering Flowchart

In this study, the data gathering would be shown in different phases. The first phase is the collecting phase, where images are gathered and collected from the object by the camera. This may have different parameters that may affect the data that would be collected, such as distance, camera orientation, stock orientation and other parameters. The next phase of data gathering is the processing of the data collected by the camera. In this phase, the picture is converted and processed to a much more compatible format of data. The final phase of the data gathering is the labeling process. In this phase, each converted and processed images are labeled and has annotations for each image and is saved in the dataset for the machine.

3.2.2 Possible Variables that are varied between collected images

Table 3.2 Parameters Possible Combinations

CAMERA DISTANCE	LIGHTING CONDITION	ORIENTATION (X-AXIS)	ROTATION (Y-AXIS)	IMAGES
3FT	5W	Normal (0°)	0°	IMAGE1
3FT	5W	Normal (0°)	45°	IMAGE2
3FT	5W	Normal (0°)	90°	IMAGE3
3FT	5W	Normal (0°)	135°	IMAGE4
3FT	5W	Normal (0°)	180°	IMAGE5
3FT	5W	Normal (0°)	225°	IMAGE6
3FT	5W	Normal (0°)	270°	IMAGE7
3FT	5W	Normal (0°)	315°	IMAGE8
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
4FT	7W	Upside-Down (180°)	0°	IMAGE105
4FT	7W	Upside-Down (180°)	45°	IMAGE106
4FT	7W	Upside-Down (180°)	90°	IMAGE107
4FT	7W	Upside-Down (180°)	135°	IMAGE108
4FT	7W	Upside-Down (180°)	180°	IMAGE109
4FT	7W	Upside-Down (180°)	225°	IMAGE110
4FT	7W	Upside-Down (180°)	270°	IMAGE111
4FT	7W	Upside-Down (180°)	315°	IMAGE112
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
5FT	12W	Sideview (90° from x-axis)	0°	IMAGE209
5FT	12W	Sideview (90° from x-axis)	45°	IMAGE210
5FT	12W	Sideview (90° from x-axis)	90°	IMAGE211
5FT	12W	Sideview (90° from x-axis)	135°	IMAGE212
5FT	12W	Sideview (90° from x-axis)	180°	IMAGE213
5FT	12W	Sideview (90° from x-axis)	225°	IMAGE214
5FT	12W	Sideview (90° from x-axis)	270°	IMAGE215
5FT	12W	Sideview (90° from x-axis)	315°	IMAGE216

Table 3.2 lists all of the possible unique raw images that will be captured and stored in the database for each object. As a result, a total of 216 images will be used for training to classify one object, resulting in a total of 432 images being created for the dataset.

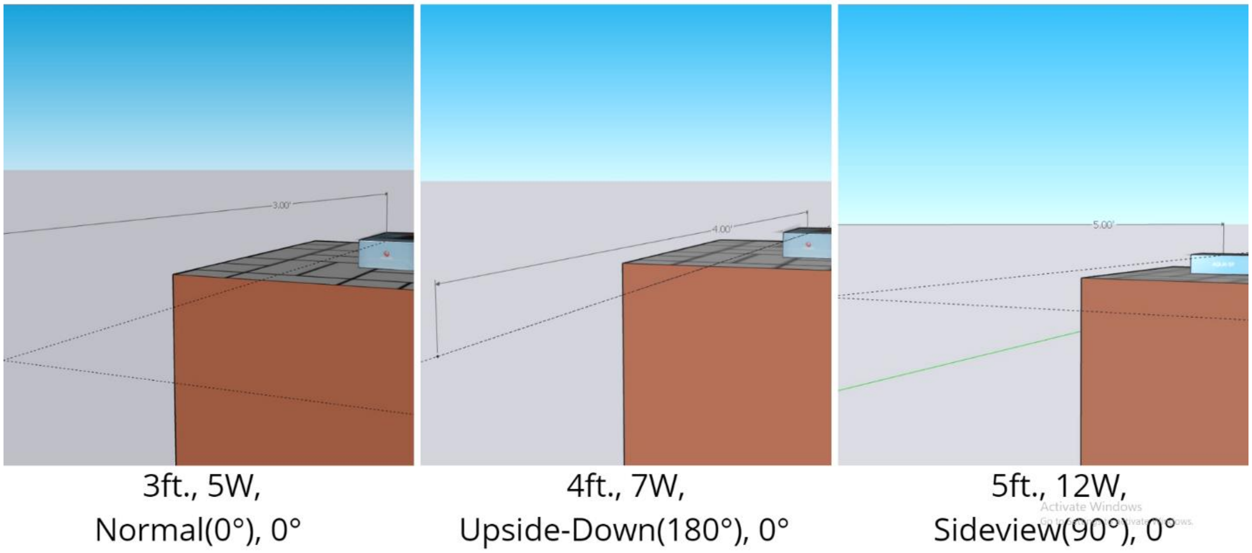


Figure 3.8 Sample Unique Raw Image in Possible Parameter Combinations

3.2.3 Dataset Collection Setup

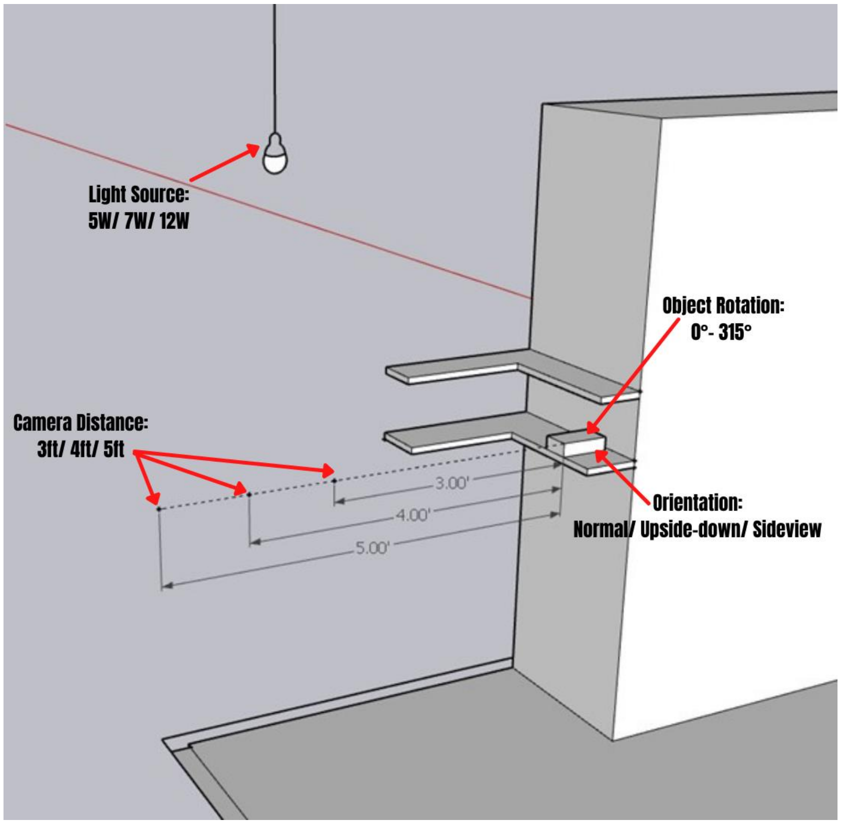


Figure 3.9 Sample Data Collection Setup

The sample data collection setup is illustrated in the design above. Each parameter that is conducting the experiments was also listed used. The light source used to simulate varying light has different wattages, such as 5 W, 7 W, and 12 W, each with its brightness. The illustration also indicates the object's orientation, which includes Normal, Upside-down, and Sideview views. Furthermore, the angle revolution for the object orientation ranges from 0 to 315 degrees relative to the object's initial position. Finally, to determine the best range for the camera to detect the object, it is placed at different distances such as 3ft, 4ft, and 5ft from the object's position.

3.2.4 Dataset Labelling & Annotation

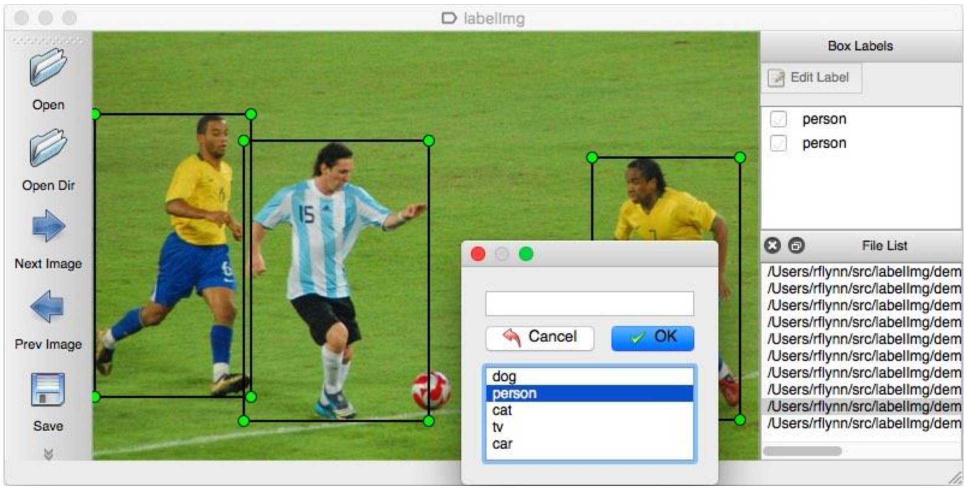

labelImg
Open
Open Dir
Next Image
Prev Image
Save
Box Labels
Edit Label
person
person
File List
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
/Users/rflynn/src/labelImg/dem
Cancel
OK
dog
person
cat
tv
car

Figure 3.10 Example of Annotating and Labeling Image using Labelmg Software

After gathering the raw image data and pre-processing it (by cropping and resizing it to a square aspect ratio), the images will be annotated with bounding boxes before being used as training data. Labeling software is used to label a sample image using bounding boxes, as shown in Figure 3.10. These annotations are then saved as xml files, which are then combined with the training data.

3.3 Validation

3.3.1 Validation Area

A validation area measuring 13 x 11 feet will be used to simulate a room scenario during validation. The camera that will be used to detect the objects will then be placed near the entrance on one side of the target object to be identified.

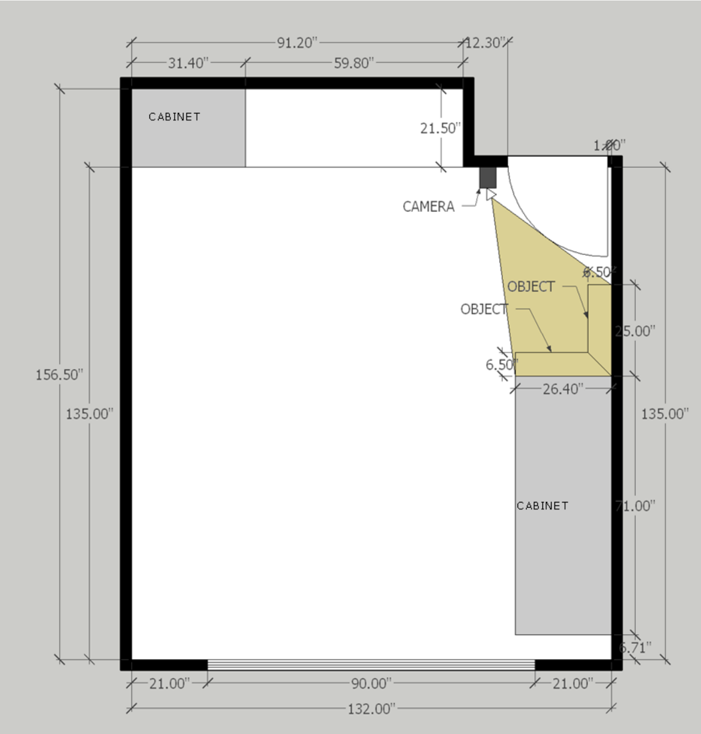
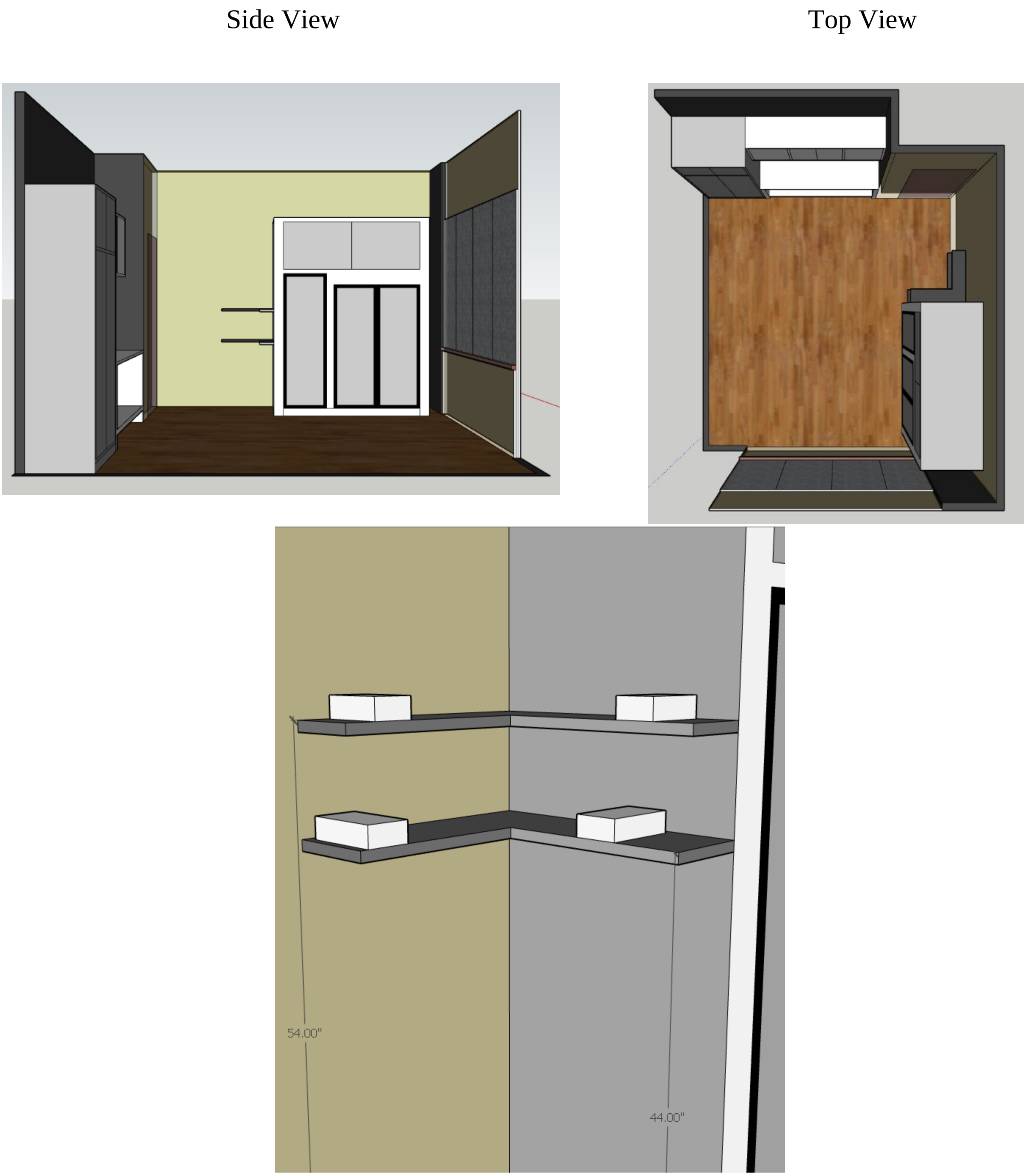


Figure 3.11 Validation Area Dimension with Camera Module for Object Detection

As shown in figure 3.11, the camera is positioned so that it can detect the target objects. The camera's maximum distance from the object is 5 ft, while the minimum is 3 ft. The camera's point of view in relation to its location is indicated by the yellow quadrilateral shade.



Side View

Top View

Figure 3.12 Perspectives of the validation area

The side and top view of the validation room is shown in Figure 3.12. The first object (Flare S8) is 4.5 feet off the ground, while the second object (Aqua S9) is 3.67 feet away.

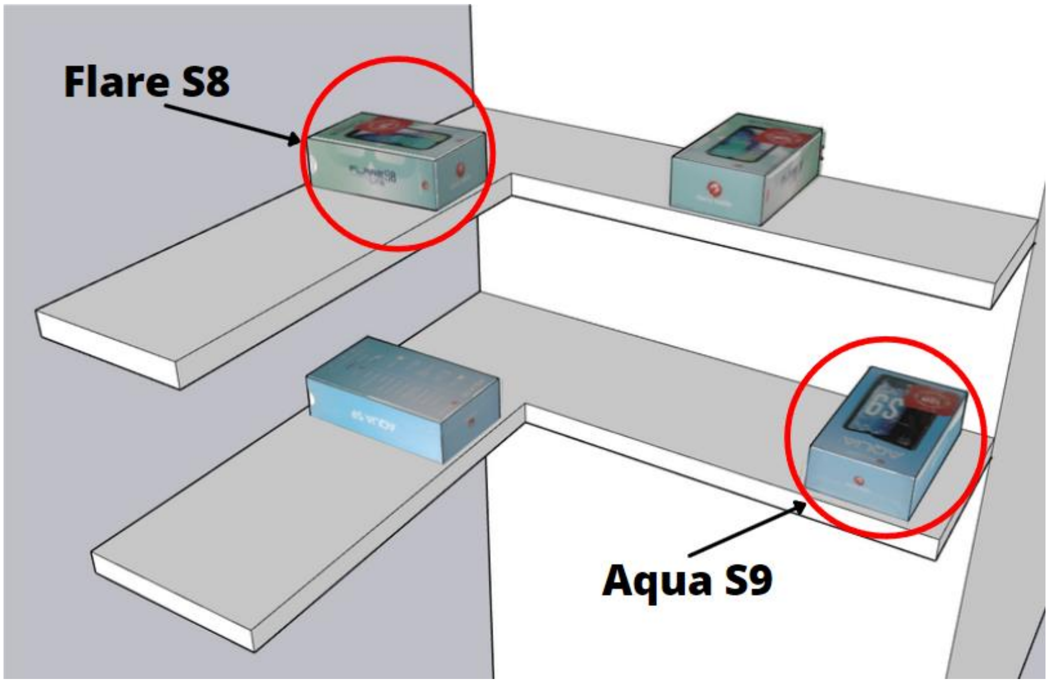


Figure 3.13 Object Position for Validation

The objects are assigned to their appropriate levels. For which the best position will recognize both objects at the same time. The first object (Aqua S9) is placed on the first level of the shelve, while the Flare S8 is placed on the second level of the shelve as shown in figure 3.13.

3.3.2 System Flowchart

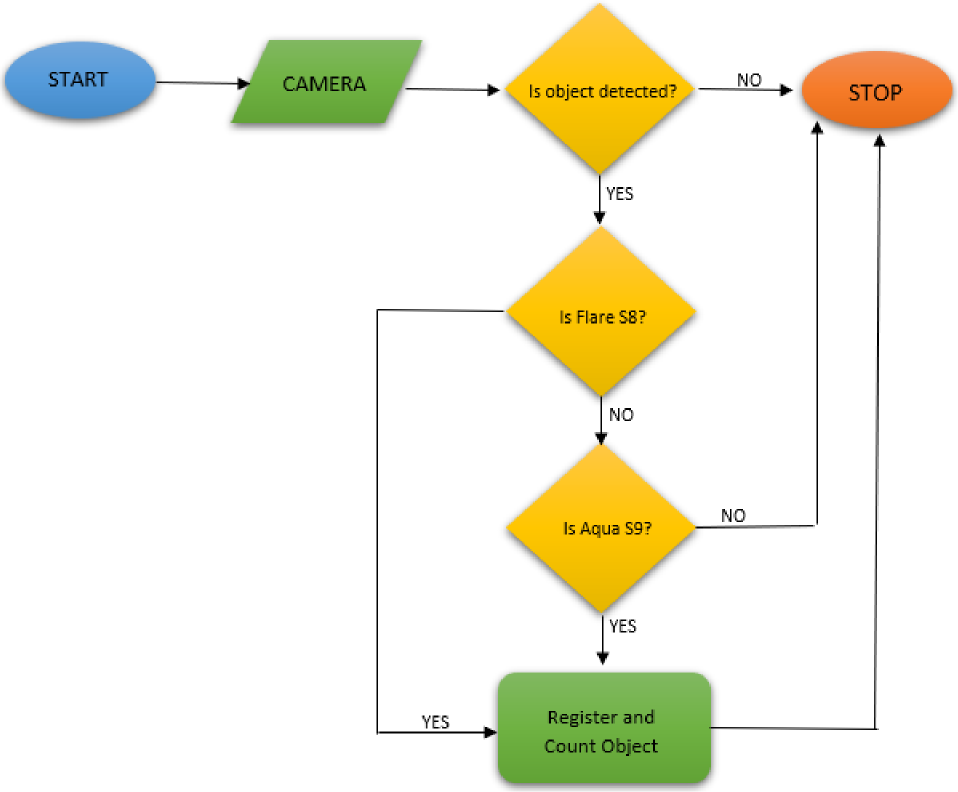
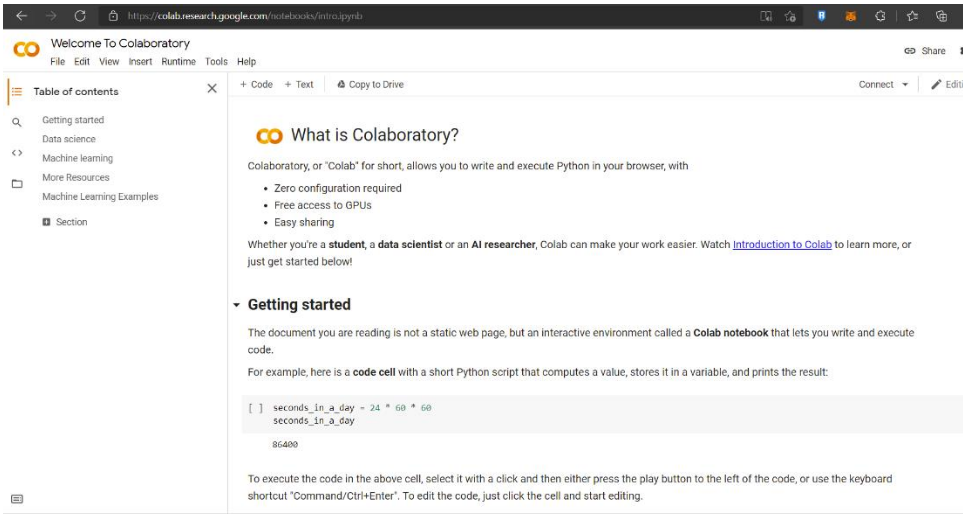


Figure 3.14 System Flowchart

During validation, this flowchart illustrates the object detection system. The camera will first determine whether there is an object within its range. If no object is found, the check will be repeated until one is found. Once an object has been detected, the algorithm will compare it to the characteristics of the two objects for which the algorithm was trained, namely the Cherry Mobile Model Flare S8 and the Aqua S9 Cellphone box. It will register and count that specific object if it matches one of the items. Otherwise, it will ignore it and continue scanning for objects.

3.4 Research Instruments



What is Colaboratory?

Colaboratory, or "Colab" for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

For example, here is a **code cell** with a short Python script that computes a value, stores it in a variable, and prints the result:

```
[ ]  seconds_in_a_day = 24 * 60 * 60
     seconds_in_a_day

     86400
```

To execute the code in the above cell, select it with a click and then either press the play button to the left of the code, or use the keyboard shortcut "Command/Ctrl+Enter". To edit the code, just click the cell and start editing.

Figure 3.15 Google Colab

In this study, Google Collab is a full-fledged cloud-based Jupyter notebook environment. The object detection algorithm model that will be used to categorize the stocks will be trained on the Google Collab Free tier, which allows up to 12 hours of free access to an NVIDIA K80.

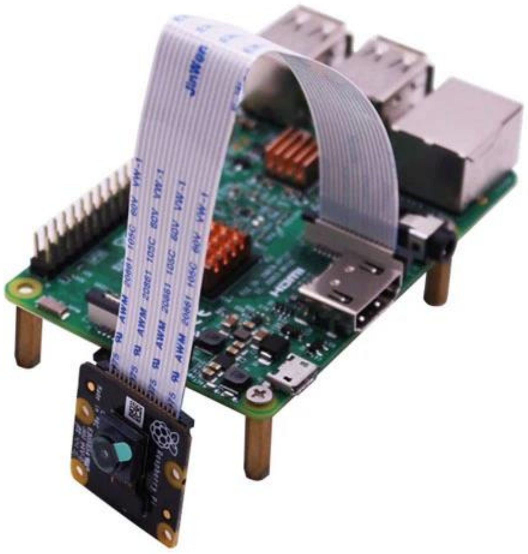
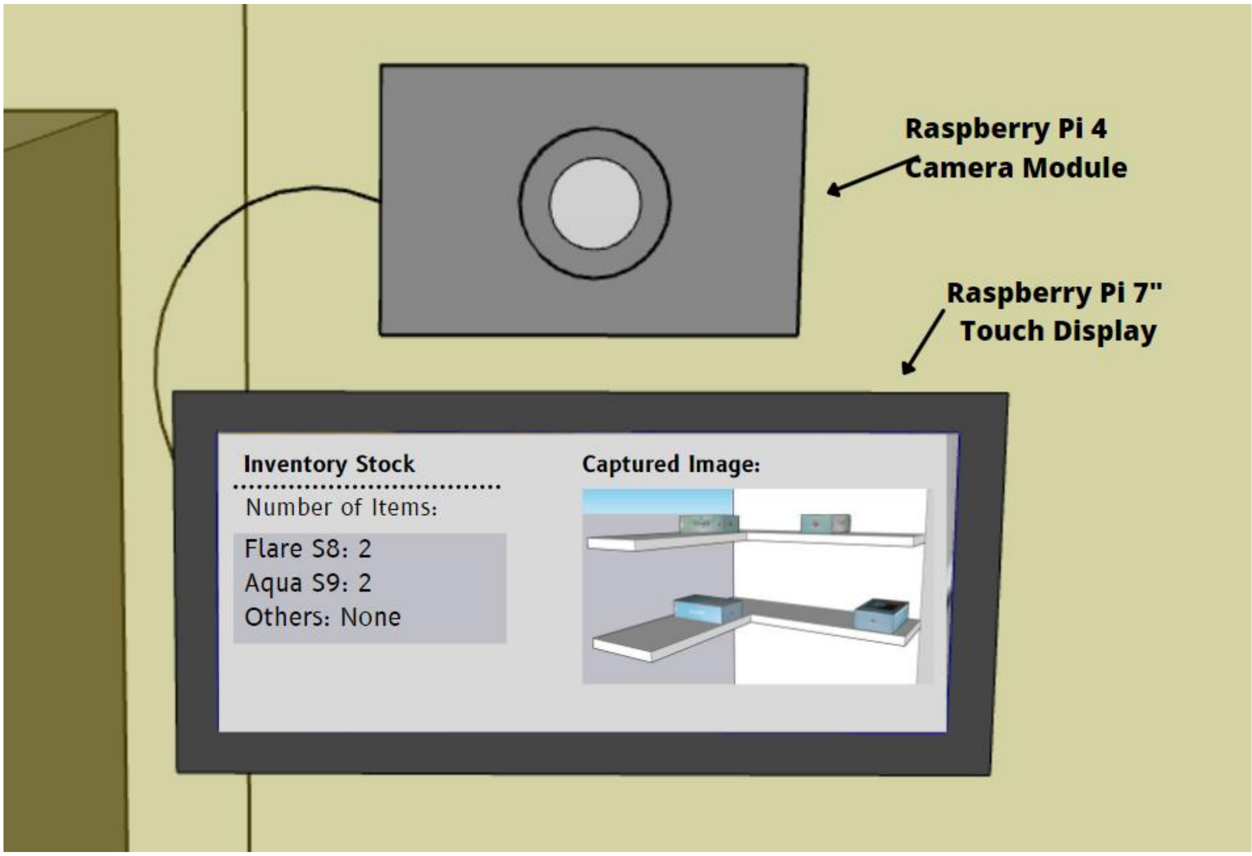


Figure 3.16 Raspberry Pi 4 and Camera Module V2

The research makes use of a camera module designed towards Raspberry Pi applications. The output video from this Camera Module V2 has a resolution of 1080p. It is a plug-and-play module that is compatible with the latest frame-ware of the Raspbian Operating System (OS). It is commonly used in a variety of applications such as time-lapse photography, video recording, motion detection, and other computer vision-related applications. A ribbon cable connects it to a Camera Serial Interface (CSI) connector.

Furthermore, the module's dimensions are approximately 25 by 23 by 99 mm, with a weight of 3 grams, a native resolution of 8 MP (megapixels), the ability to capture images of 3280 by 2464 pixels, and the ability to record video in 1080 pixels with 30 frames, 720 pixels with 60 frames, and 640 by 480 pixels with 90 frames.



Number of Items:

Flare S8: 2
Aqua S9: 2
Others: None

Figure 3.17 Proposed Camera and Touch Display Setup Showing a Sample Designed Application

The object detection algorithms will be run on the Raspberry Pi 4 Model B after it has been trained in Google Collab. The Raspberry Pi 4 Model B has a Quad Core Cortex-A72 64-bit SoC running at 1.5GHz with 4GB of LPDDR4-3200 SDRAM, making it ideal for object categorization applications. Afterwards, a Raspberry Pi Camera Module V2 will be utilized to capture camera as an input to the trained algorithm. An application will be designed and programmed in such a way that it can link with the microcontroller and monitor what the system has detected and recognized. The application will detect the number of items automatically and display the count on the Raspberry Pi 7” Touch Display. A pre-captured image will appear in the GUI as seen from figure 3.17.

3.5 Region of Interest Pooling

Region of interest pooling also known as RoI pooling is a common operation in convolutional neural network object detection tasks. Its goal is to perform max pooling on nonuniformly sized inputs to obtain fixed-size feature maps.

To see how Region of Interest (RoI) works, Figure 3.18 shows a single 9×6 feature map, one region of interest and an output size of 2×2. Then, performing region of interest pooling on the input feature map. Figure 3.19 shows the region proposal. There would normally be a multiple feature maps and proposals for each. Figure 3.20 shows dividing regional proposal in (2×2) sections, because the output size is 2×2. The size of the region of interest does not have to be exactly divided by the number of pooling sections. In this case, The RoI is 7×4 and having a 2×2 pooling sections. [34]

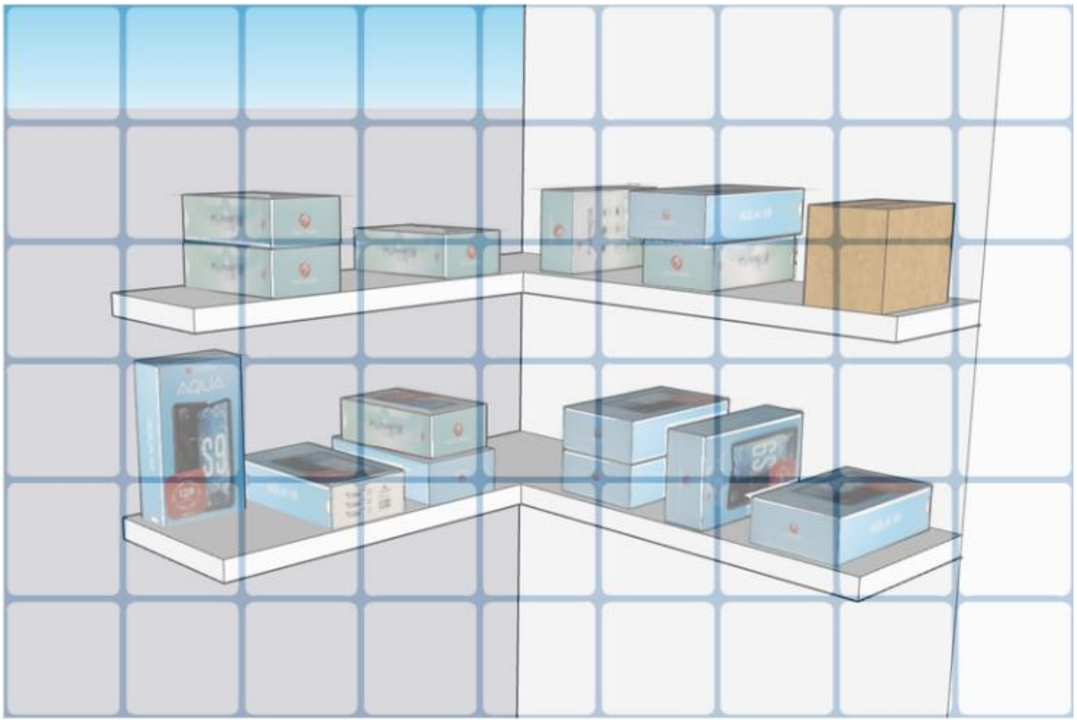


Figure 3.18 Input Feature Map

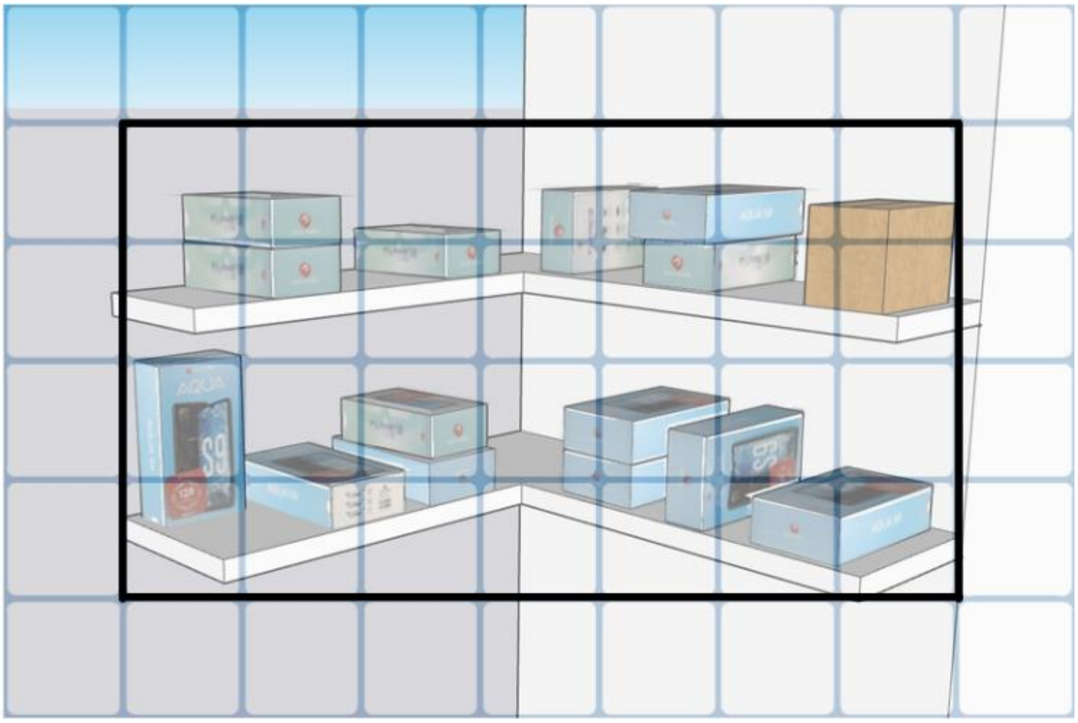


Figure 3.19 Region Proposal

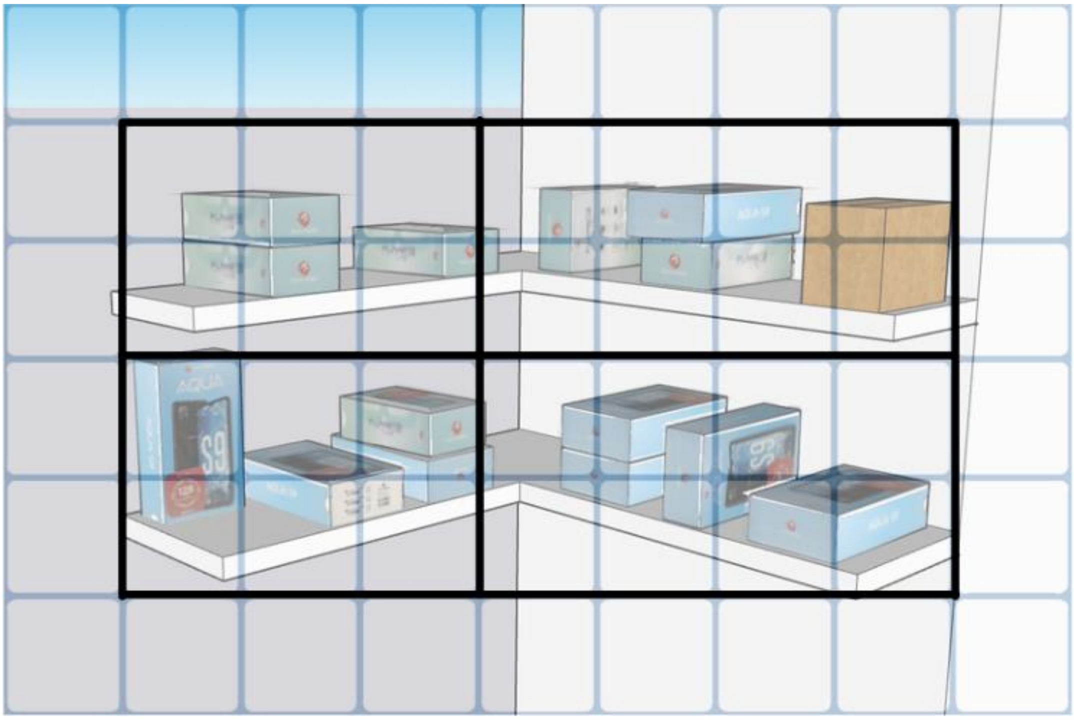


Figure 3.20 Pooling Sections

3.6 Results

3.6.1 Actual and Prediction

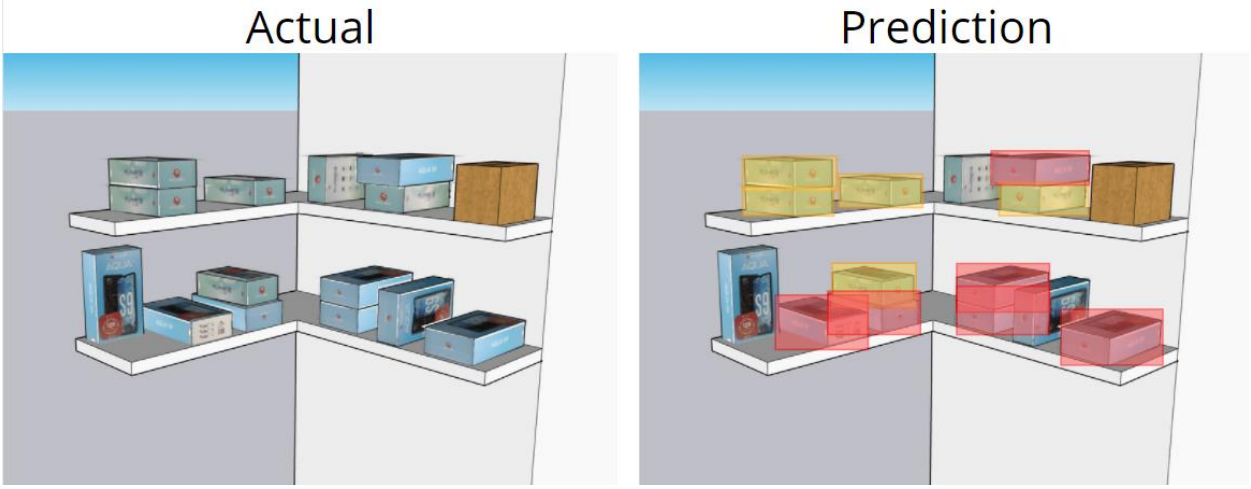


Figure 3.21 Sample Actual vs Prediction on Object Detection and Counting

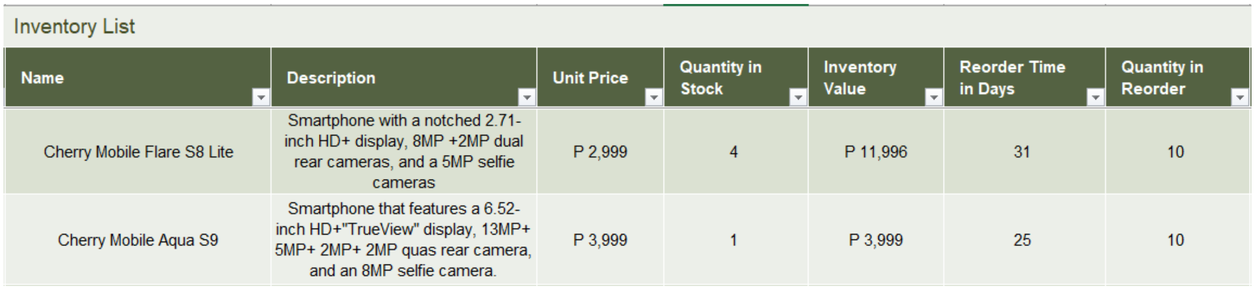
Figure 3.21 shows the actual and the prediction on object detection and counting. The first image is the actual and the second image is the prediction. As seen from the first image there is a total of fifteen objects on the shelves, on the other hand, for the prediction, the highlighted objects are the only objects that will be detected. There are four objects that were not detected because either there is another object (cardboard box) or there is a different object orientation in the image.

Table 3.3 Trial Sample (Based on Figure 3.21)

Object predicted	**Orientation**	**Number of Objects Detected**	**Are there Objects Detected?**
Flare S8	Normal	1	Yes
Flare S8	Upside-Down	1	Yes
Flare S8	Sideways	1	No
Flare S8	Upright	1	No
Aqua S9	Normal	1	Yes
Aqua S9	Upside-Down	1	Yes
Aqua S9	Sideways	1	No
Aqua S9	Upright	1	No
Flare S8 and Flare S8	Normal	2	Yes
Aqua S9 and Aqua S9	Normal	2	Yes
Aqua S9 and Flare S8	Normal	2	Yes
Flare S8 and Aqua S9	Normal	2	Yes
With Noise	Normal	3	Yes
With Noise	Upside-Down	4	Yes
With Noise	Sideways	5	No
With Noise	Upright	6	No
No items	Normal	0	No
No items	Normal	0	No
No items	Normal	0	No
No items	Normal	0	No

3.6.2 Stock Control

Table 3.4 Inventory Stock Spreadsheet



Inventory List

Name	Description	Unit Price	Quantity in Stock	Inventory Value	Reorder Time in Days	Quantity in Reorder
Cherry Mobile Flare S8 Lite	Smartphone with a notched 2.71-inch HD+ display, 8MP +2MP dual rear cameras, and a 5MP selfie cameras	P 2,999	4	P 11,996	31	10
Cherry Mobile Aqua S9	Smartphone that features a 6.52-inch HD+"TrueView" display, 13MP+ 5MP+ 2MP+ 2MP quas rear camera, and an 8MP selfie camera.	P 3,999	1	P 3,999	25	10

As seen from table 3.4, the stocks will be monitored Monthly. Spreadsheets are an excellent way to track inventory, automating and electronically capture product data, whether using a Microsoft Excel or something similar. Ensuring having a current inventory levels and statistics by updating regularly and using basic coding.

3.6.3 Accuracy Test

Table 3.5 Accuracy Test for Object Detection

PREDICTED	**Cherry Mobile Flare S8**	**Cherry Mobile Aqua S9**	**Noise**	**None**
Flare S8	TA	FA1	FA2	FA3
Aqua S9	FB1	TB	FB2	FB3
Noise (Background Noise or other items)	FC1	FC2	TC	FC3

None	FD1	FD2	FD3	TD

Table 3.6 Accuracy Test for Object Count

Object	**Prediction**	**Manual**	**Accuracy %**
Flare S8			
Aqua S9			
		Average Accuracy:	

Table 3.5 and table 3.6 will be used to determine the trained model's accuracy. The dataset will also be split, with 70% of it being used to train the algorithm and 30% being used to test it. The confusion matrix will be built using data from the test dataset.

The accuracy of a model refers to its ability to correctly classify data. The average image detection accuracy. The accuracy equation can be used to calculate the accuracy value.

Where:

$$Accuracy = \frac{TP + TN}{N}$$

TP = Number of cases where the prediction is Flare S8/ Aqua S9, and the predicted is also Flare S8/ Aqua S9.

TN = Number of cases where the prediction is not Flare S8/ Aqua S9, and the predicted is also not Flare S8/ Aqua S9.

N = The total number of cases

REFERENCES

[1] “Stock control and inventory“ Info [Online serial]. Available: https://www.infoentrepreneurs.org/en/guides/stock-control-and-inventory/ [Accessed Sept 6, 2021]

[2] S. Armalivia, Z. Zainuddin, A. Achmad and M. A. Wicaksono, "Automatic Counting Shrimp Larvae Based You Only Look Once (YOLO)," 2021 International Conference on Artificial Intelligence and Mechatronics Systems (AIMS), 2021, pp. 1-4, doi: 10.1109/AIMS52415.2021.9466058.

[3] L. Hulstaert, “Object Detection Tutorial with SSD & Faster RCNN” DataCamp Community, April 19, 2018. [Online serial]. Available: https://www.datacamp.com/community/tutorials/object-detection-guide [Accessed Sept 6, 2021]

[4] K.R. Velasco “YOLO (You Only Look Once)” towards data science, June 06, 2019. [Online serial]. Available: https://towardsdatascience.com/yolo-you-only-look-once-17f9280a47b0 [Accessed Sept 6, 2021]

[5] J. Lin and M. Sun, "A YOLO-Based Traffic Counting System," 2018 Conference on Technologies and Applications of Artificial Intelligence (TAAI), 2018, pp. 82-85, doi: 10.1109/TAAI.2018.00027.

[6] P. Banerjee, “YOLO—You only look once” towards data science, May 30, 2020. [Online serial]. Available: https://towardsdatascience.com/yolo-you-only-look-once-3dbdbb608ec4 [Accessed Sept 15, 2021]

[7] Fathoni, F. A., Ridwan, A. Y., & Santosa, B. (2018, November). Development of inventory control application for pharmaceutical product using abc-ved cycle counting method to increase inventory record accuracy. In *Proceedings of the 2018 International Conference on Industrial Enterprise and System Engineering (IcoIESE), Yogyakarta, Indonesia* (pp. 21-22). https://www.researchgate.net/profile/Ari-Ridwan/publication/332226370_Development_of_Inventory_Control_Application_for_Pharmaceutical_Product_Using_ABC-VED_Cycle_Counting_Method_to_Increase_Inventory_Record_Accuracy/links/5ff0235b45851553a0110175/Development-of-Inventory-Control-Application-for-Pharmaceutical-Product-Using-ABC-VED-Cycle-Counting-Method-to-Increase-Inventory-Record-Accuracy.pdf [Accessed October 2, 2021].

[8] Casamayor-Pujol, V., Morenza-Cinos, M., Gastón, B., & Pous, R. (2020). Autonomous stock counting based on a stigmergic algorithm for multi-robot systems. *Computers in Industry*, *122*, 103259. https://www.sciencedirect.com/science/article/abs/pii/S0166361520304930. [Accessed October 2, 2021].

[9] Verma, N. K., Sharma, T., Sevakula, R. K., & Salour, A. (2016, December). Vision based object counting using speeded up robust features for inventory control. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 709-714). IEEE. https://ieeexplore.ieee.org/abstract/document/7881432. [Accessed October 2, 2021].

[10] Balderas-López, E. M. (2020). Warehouse stock counting prototype using Raspberry Pi and OpenCV.

https://rei.iteso.mx/bitstream/handle/11117/6549/TOG%20Especialidad%20Sistemas%20Embebidos%20Edwin%20Balderas.pdf?sequence=1&isAllowed=y. [Accessed October 2, 2021].

[11] Zhiqiang, W., & Jun, L. (2017, July). A review of object detection based on convolutional neural network. In *2017 36th Chinese Control Conference (CCC)* (pp. 11104-11109). IEEE. https://ieeexplore.ieee.org/abstract/document/8029130. [Accessed October 2, 2021].

[12] Dhillon, A., & Verma, G. K. (2020). Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, *9*(2), 85-112. https://link.springer.com/article/10.1007/s13748-019-00203-0. [Accessed October 2, 2021].

[13] Galvez, R. L., Bandala, A. A., Dadios, E. P., Vicerra, R. R. P., & Maningo, J. M. Z. (2018, October). Object detection using convolutional neural networks. In *TENCON 2018-2018 IEEE Region 10 Conference* (pp. 2023-2027). IEEE. https://ieeexplore.ieee.org/abstract/document/8650517. [Accessed October 2, 2021].

[14] Kang, K., Li, H., Yan, J., Zeng, X., Yang, B., Xiao, T., ... & Ouyang, W. (2017). T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, *28*(10), 2896-2907. https://ieeexplore.ieee.org/abstract/document/8003302. [Accessed October 3, 2021].

[15] Zhang, Y., Yuan, Y., Feng, Y., & Lu, X. (2019). Hierarchical and robust convolutional neural network for very high-resolution remote sensing object detection. *IEEE*

Transactions on Geoscience and Remote Sensing, *57*(8), 5535-5548. https://ieeexplore.ieee.org/abstract/document/8676107. [Accessed October 3, 2021].

[16] Hurtik, P., Molek, V., & Vlasanek, P. (2020, July). YOLO-ASC: you only look once and see contours. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE. https://ieeexplore.ieee.org/abstract/document/9207223. [Accessed October 3, 2021].

[17] Adibhatla, V. A., Chih, H. C., Hsu, C. C., Cheng, J., Abbod, M. F., & Shieh, J. S. (2020). Defect detection in printed circuit boards using you-only-look-once convolutional neural networks. *Electronics*, *9*(9), 1547. https://www.mdpi.com/2079-9292/9/9/1547. [Accessed October 3, 2021].

[18] Shafiee, M. J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *arXiv preprint arXiv:1709.05943*. https://arxiv.org/abs/1709.05943. [Accessed October 3, 2021].

[19] 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). https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html. [Accessed October 3, 2021].

[20] Cao, C. Y., Zheng, J. C., Huang, Y. Q., Liu, J., & Yang, C. F. (2019). Investigation of a promoted you only look once algorithm and its application in traffic flow monitoring. *Applied Sciences*, *9*(17), 3619. https://www.mdpi.com/2076-3417/9/17/3619/htm. [Accessed October 3, 2021].

[21] Xiang, W., Zhang, D. Q., Yu, H., & Athitsos, V. (2018, March). Context-aware single-shot detector. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1784-1793). IEEE. https://ieeexplore.ieee.org/abstract/document/8354302. [Accessed October 3, 2021].

[22] Biswas, D., Su, H., Wang, C., Stevanovic, A., & Wang, W. (2019). An automatic traffic density estimation using Single Shot Detection (SSD) and MobileNet-SSD. *Physics and Chemistry of the Earth, Parts A/B/C, 110,* 176-184. https://www.sciencedirect.com/science/article/pii/S1474706518302389. [Accessed October 3, 2021].

[23] Li, Z., & Zhou, F. (2017). FSSD: feature fusion single shot multibox detector. *arXiv preprint arXiv:1712.00960*. https://arxiv.org/abs/1712.00960. [Accessed October 3, 2021].

[24] Chen, W., Qiao, Y., & Li, Y. (2020). Inception-SSD: An improved single shot detector for vehicle detection. *Journal of Ambient Intelligence and Humanized Computing*, 1-7. https://link.springer.com/article/10.1007/s12652-020-02085-w. [Accessed October 4, 2021].

[25] Ning, C., Zhou, H., Song, Y., & Tang, J. (2017, July). Inception single shot multibox detector for object detection. In *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 549-554). IEEE. https://ieeexplore.ieee.org/abstract/document/8026312. [Accessed October 4, 2021].

[26] Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018, May). Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. In

2018 15th Conference on Computer and Robot Vision (CRV) (pp. 95-101). IEEE. https://ieeexplore.ieee.org/abstract/document/8575741. [Accessed October 4, 2021].

[27] Luvizon, D., Tabia, H., & Picard, D. (2020). SSP-Net: Scalable Sequential Pyramid Networks for Real-Time 3D Human Pose Regression. *arXiv preprint arXiv:2009.01998*. https://arxiv.org/abs/2009.01998. [Accessed October 5, 2021].

[28] Zou, F., Xiao, W., Ji, W., He, K., Yang, Z., Song, J., ... & Li, K. (2020). Arbitrary-oriented object detection via dense feature fusion and attention model for remote sensing super-resolution image. *Neural Computing and Applications*, *32*(18), 14549-14562. https://www.mdpi.com/2072-4292/12/2/339. [Accessed October 5, 2021].

[29] Yang, X., Sun, H., Fu, K., Yang, J., Sun, X., Yan, M., & Guo, Z. (2018). Automatic ship detection in remote sensing images from google earth of complex scenes based on multiscale rotation dense feature pyramid networks. *Remote Sensing*, *10*(1), 132. https://www.mdpi.com/2072-4292/10/1/132. [Accessed October 5, 2021].

[30] Hu, G. X., Yang, Z., Hu, L., Huang, L., & Han, J. M. (2018). Small object detection with multiscale features. *International Journal of Digital Multimedia Broadcasting*, *2018*. https://www.hindawi.com/journals/ijdmb/2018/4546896/. [Accessed October 5, 2021].

[31] G. Maindola. (June 20, 2021). Introduction to YOLOv5 Object Detection with Tutorial. *MLK - Machine Learning* Knowledge, https://machinelearningknowledge.ai/introduction-to-yolov5-object-detection-with-tutorial/. [Accessed October 13, 2021].

[32] Redmon, J. (2012). YOLO: Real-Time Object Detection. Pjreddie.com. [Online serial]. Available: https://pjreddie.com/darknet/yolo/ [Accessed Oct 9, 2021].

[33] Nath, Nipun & Behzadan, Amir. (2020). Deep Convolutional Networks for Construction Object Detection Under Different Visual Conditions. Frontiers in Built Environment. 6. 10.3389/fbuil.2020.00097. [Accessed Oct 9, 2021].

[34] T. Grel. (2017). Region of interest pooling explained. Deepsense.ai [Online serial]. Available: https://deepsense.ai/region-of-interest-pooling-explained/ [Accessed Oct 15, 2021].