A

Project Report

on

Automated Retail Checkout System using YOLOv5 in Computer Vision

submitted for partial fulfillment for the award of

BACHELOR OF TECHNOLOGY

DEGREE

in

Computer Science

By

KSHITEESH KUMAR (2000290120088)
KUMARI BHAVYA CHAUBEY (2000290120089)
NANDITA YADAV (2000290120100)

Under the Supervision of

Mr. Anmol Jain
Assistant Professor

Department of Computer Science
KIET Group of Institutions, Ghaziabad
Affiliated to
Dr. A.P.J. Abdul Kalam Technical University, Lucknow

May 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Kshiteesh Kumar (2000290120088)

Kumari Bhavya Chaubey (2000290120089)

Nandita Yadav (2000290120100)

Date:

CERTIFICATE

This is to certify that Project Report entitled **"Automated Retail Checkout using YOLOv5 in Computer Vision"** which is submitted by **Kshiteesh Kumar, Kumari Bhavya Chaubey, Nandita Yadav** in partial fulfillment of the requirement for the award of degree.

B. Tech. in the Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates' own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Date: **Supervisor Signature:**

Mr. Anmol Jain

Assistant Professor

Department of

ComputerScience

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe a special debt of gratitude to **Mr. Anmol Jain**, Assistant Professor, Department of Computer Science, KIET, Ghaziabad, for his constant support and guidance throughout our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen the light of the day.

We also acknowledge the contribution of **Dr. Ajay Kumar Shrivastava**, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the project's development. We would not want to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperationduring the development of our project.

Finally, we acknowledge our families and friends for their contribution to the completion of the project.

Name	**Roll No.**	**Signature**
Kshiteesh Kumar	2000290120088	
Kumari Bhavya Chaubey	2000290120089	
Nandita Yadav	2000290120100	

TABLE OF CONTENTS

ABSTRACT

There have been rapid advancements in the field of technology since the arrival of computers and internet. With the advent of technology, the retail sector has witnessed significant changes that once used to be believed too unfathomable. Self-service checkout systems have been established in a variety of stores in which customers have to scan all the bought items with a barcode reader and then pay for the bought items. However, these services could result in thefts in these stores and reduce customer experience and impersonal communication between employees and customers. Technologies such as Amazon Go, released by Amazon in 2017, which is based on "Just Walk Out Technology", eased the self-checkout process as the customers do not have to wait in queues to pay for the purchased items. After installing the Amazon Go app, customers scan their iPhones as they enter the store. The track of their purchases is maintained. They can leave the store without physically checking out as their accounts are charged automatically. It makes use of deep learning, computer vision, and sensors. It is built on technology like self-driving cars. However, this technology can hinder customer experience as customers unfamiliar with the app would find it very difficult to shop. Besides the technology used in apps like Amazon Go is expensive.

Therefore, to solve these problems and produce an outcome that could favor both the customer, our implementation focuses on enhancing customer experience by creating a system that can scan the bought items, detect these items with the help of a pretrained model, and generate a bill containing the total cost that needs to be paid.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

GUI	Graphic User Interface
YOLO	You Only Look Once
COCO	Common Objects in Context
CSV	Comma Separated Value
YAML	Yet Another Markup Language

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays automation techniques have proved to be a lot beneficial in the retail sector. They play a crucial role in the retail sector by providing better efficiency, reducing costs, enhancing customer experience as well as giving a competitive edge in today's fast-paced market. Many applications make use of Computer Vision techniques to simulate the behavior of human eyesight.

Our system is based on the YOLOv5 technique of object detection. It is a model in the YOLO series of Computer Vision models and is used to detect multiple objects in an image with fast speed and better accuracy. Our implementation is an automated retail checkout system which generates the bill of the bought products by taking images or videos of the bought products. The main objective of this system is to ease the process of checkout at the retail stores. It can help to reduce the long queues at the checkout counters and can therefore, save customers' time. For pre-processing the dataset, Roboflow is used. Roboflow has an inbuilt feature that can be used to give annotations. Using it, we put labels on each of the photos. Along with this, Roboflow increases the dataset by rotating every image with different angles. After processing the dataset from Roboflow, we import data in YOLOv5 for further detection. Object Detection and Object Recognition are performed through YOLOv5 which is trained on the COCO Dataset. YOLOv5 is used to achieve better accuracy in object detection. It is trained on custom datasets. The system can then generate an automated bill that includes the information like name of the product, its quantity, and its price. The final amount of all the bought items is also written at the end of the report.

1.2 Project Category

Our project belongs to the **machine learning based** category. It automates the process of checkouts at the retail stores and thus helping in easing the process of checkouts at the retail stores. Specifically, it involves object detection and recognition, which is a common application of computer vision technology. Computer Vision is the overarching field dealing with machines interpreting and understanding the visual world, while Automated Retail Systems refer to technologies and systems aimed at automating various processes within retail environments.

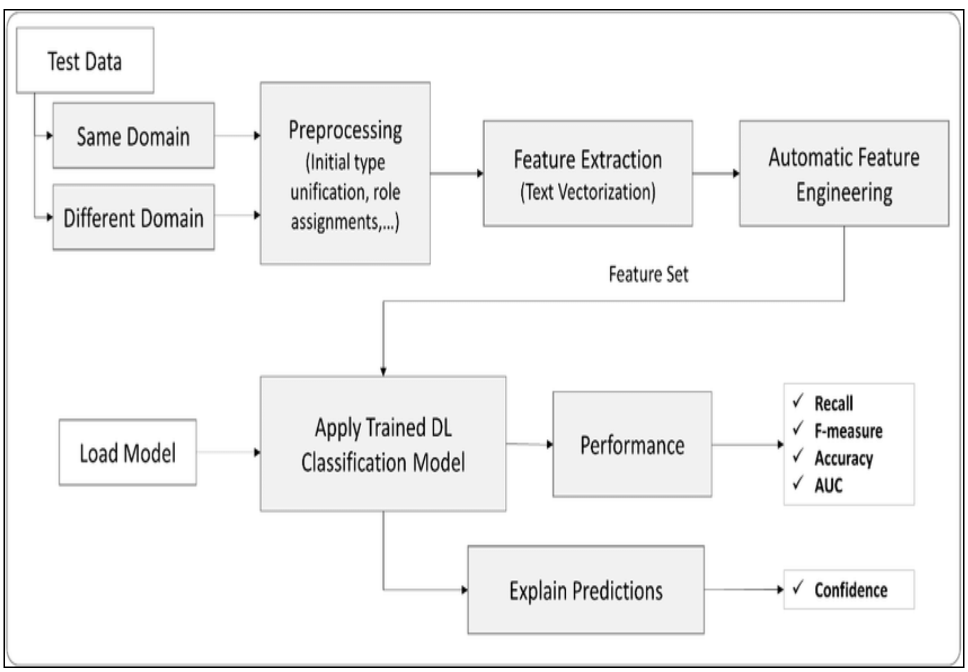


Fig. 1.1. Process of Deep Learning

1.3 Objectives

The objectives of the system typically include:

1.3.1 Efficiency

To automate the checkout process to reduce waiting times for customers, leading to improved customer satisfaction.

1.3.2 Accuracy

Ensuring accurate identification and detection of items being purchased, minimizing errors in the checkout process and reducing the need for manual intervention.

1.3.3 Loss Prevention

Identifying instances of theft or fraud by comparing items detected by the system with those registered for purchase, and flagging discrepancies for further investigation.

1.3.4 Customer Experience

Enhancing the overall shopping experience by streamlining the checkout process, reducing queues, and providing a more convenient and seamless transaction process for customers.

1.3.5 Adaptability

Building a system that can adapt to different retail environments, handle various types of products, and accommodate changes in store layout or configuration.

1.4 Structure of Report

The project report has been divided into multiple chapters with each chapters getting divided into multiple sections and sub-sections.

Chapter 1 contains the introduction where a brief introduction of the project, the category to which the project belongs to and the objectives that the system proposes to achieve, are mentioned. In Chapter 2, we have discussed the previous research work that has happened in this field as well as the research gaps and the problem statement. Chapter 3 contains the detailed discussion of the project, its functionalities, the methodology used to create the proposed system and the unique features of the project.

In chapter 4, we have performed the feasibility study and mentioned the various parameters and evaluated the system's technical, economical and operational feasibility. Along with this, we have also included the software requirement specification document and Software Development Life Cycle (SDLC) model that has been used. It also illustrates the various diagrams like System Design using DFD Level 0 and Level 1, Use Case Diagram and ER Diagram.

Chapter 5 includes the implementation details of the proposed system. It mentions the languages, tools and technologies that have been used to create the system whereas chapter 6 is dedicated to the testing and maintenance of the project.

In Chapter 7, the results and discussion is done that contains the user interface representation, description of various modules of the project and snapshots of the system. The final chapter 8 is dedicated to the conclusion and future scope of the system.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review

Many researchers have investigated the applications of Computer Vision in managing retail stores. They have successfully carried out studies and explained the methodologies that could help the retailers manage these stores. A brief literature review is explained in this section.

In 2019, D. A. Mora Hernandez, O. Nalbach, and D. Werth introduced a conceptual tracking system that could generate movement tracks over time for individual customers. In 2020, N. Shekokar, A. Kasat, S. Jain, P. Naringrekar, and M. Shah, prepared an innovative model 'Shop and Go' which makes use of deep learning and sensor fusion. Sensors were used to detect whether the weight had been changed if the item was picked or not. They also stated some real-time applications along with their pros and cons. The real-time applications of Self-Checkout Systems include real time recommendations, management interface, efficient staff deployment, shoplifting preventions and shop layout optimizations.

Besides there was also research conducted as to how computer vision can be used to manage product stock at offline retail stores. In 2020, M. A. Majdi, B. Sena Bayu Dewantara, and M. M. Bachtiar proposed a system to find out which goods were nearly empty and misplaced. A camera was used to capture all the displayed products and the products were recognized using YOLOv3, which resulted in achieving the accuracy of 97.61% and 76.67% for misplaced detection. In 2020, C. G. Melek, Elena Battini

Sonmez, and Songul Albayrak compared various object detection algorithms and concluded that YOLOv2 is a far better object detection algorithm in terms of both performance and speed for object detection in shelf images.

S. K. Yedla, V. M. Manikandan, and V. Panchami in 2020, proposed a novel approach for real-time scene change detection with object detection for automated stock verification. They used a scene change detection technique based on a Structural Similarity Index (SSIM) with the goal of optimizing the processing of video frames.
In the work proposed by M. Sugadev, K. Sucharitha, I. R. Sheeba, and B. Velan, the design and implementation of a hardware-based automated billing system aimed for fruit shops and achieving the Granny Smith Accuracy of 98.896%, was described. It proposed more efficient and accurate billing system than traditional billing systems. They used neural network to classify the fruit and load cell to find the weight of the fruit. In 2020, H. Y. Putra proposed a fraud detection system to detect fraud at self-checkout stores using Data Mining. He predicted fraud using classification techniques and visualized the results to obtain new insight. J48 model had the best performance with F-measure 0.921.

Along with these, there have been research on various object detection algorithms and how do they differ from each other in terms of various factors. One of the steps involved in Computer Vision is Object Detection. Its main objective is to find and localize the objects present in an image or a video. It entails two tasks: Image Classification and Object Localization. While object localization involves locating objects in an image and using a bounding box to denote their locations, image classification identifies the kind or class of an object. In the case of detecting multiple objects, multiple object detection algorithms are used.

Convolutional Neural Networks (CNN) are popularly used in object detection. Over the past few years, CNN has become famous for all the tasks involved in computer vision as well as object detection.

Object detection methods that are based on CNN, are grouped into two models: One-Stage Models and Two-Stage Models. They are explained as:

One-Stage Models:

These models require one pass only through the neural network and predict the bounding boxes in one go. They prioritize inference speed. YOLO series, SSD, and RetinaNet are the representative networks. You Only Look Once (YOLO) is a widely used object detection algorithm, capable of detecting objects in real-time. It has many versions such as YOLOv3, YOLOv5, YOLOv6, etc. SSD (Single Shot Detector) pertains to multi-scale target feature extraction helping in predicting multiple classes. RetinaNet introduces a new function known as the Focal Loss Function that deals with class imbalance during training.

Two-Stage Models:

These models consist of two steps: creating multiple proposal bounding boxes in the first phase, and using these bounding boxes to perform item categorization and location information prediction in the second phase. These models prioritize detection accuracy over inference speed. R-CNN, Fast R-CNN, SPPNet, Faster R-CNN, FPN, and Mask-RCNN are the representative models of Two-Stage Models. R-CNN uses selective search approach to propose regions that likely contain objects. Instead of using a selective search algorithm, Fast R-CNN establishes a pooling layer known as Region of Interest (RoI) that extracts fixed-size feature maps from CNN feature maps whereas Faster R-CNN uses an integrated Region Proposal Network (RPN) that shares convolutional features with the detection network. Mask R-CNN extends Faster R-CNN by including a branch for predicting segmentation masks, along with bounding boxes. Spatial Pyramid Pooling Network (SPPNet) is used for arbitrary images without requiring resizing.

Markov Clustering Networks (MCN) and the Viola-Jones technique are used to detect multiple objects and the presence of faces and text within an image. However, these approaches do not account for things like picture size, gradient, orientation, and characters.

YOLOv5 uses Efficient Det, which is a more complex architecture unlike the previous versions of YOLO. This helps to getter better accuracy and improved generalization of object categories. Apart from this, YOLOv5 is trained on 600 object categories which is a more diverse dataset than other versions. It also utilizes Spatial Pyramid Pooling (SPP) which helps to improve the performance of the model in detecting the smaller objects.

The application requirements as well as several other factors, such as speed, accuracy, handling small objects, training complexity, and real-time object detection capacity, influence the choice of object detection method. YOLO excels in real-time object detection as compared to other algorithms. SSD provides a good compromise between speed and accuracy and can handle small objects efficiently.

In Table 1, one-stage object detection methods - SSD and YOLO, and two-stage object detection methods - R-CNN, Fast R-CNN, and Faster R-CNN, are compared based on various parameters like speed, real-time object detection, accuracy in detecting small objects, and training complexity. Each of these algorithms serves some purpose and is utilized according to the need.

Table 2.1. Comparison of Some Object Detection Methods

Point of Difference	**R-CNN**	**Fast R-CNN**	**Faster R-CNN**	**SSD**	**YOLO**
Speed	Slow	Medium	Medium	Fast	Fast
Real Time Object Detection	Low	Better than R-CNN	Better than R-CNN and Fast R-CNN	Best	Best
Accuracy in Detecting Small Objects	Low	Better than R-CNN	Better than YOLO	High	Low
Training Complexity	High	Easier than R-CNN	Easier than R-CNN and Fast R-CNN	Easier than two-stage detection methods	Low

2.2 Research Gaps

Automated retail checkout systems have become increasingly popular in recent years. However, there are still research gaps and challenges in the development and implementation of these systems:

2.2.1 Fraud Prevention:

Automated retail checkout systems are vulnerable to theft and fraud. Research is needed to develop more advanced security measures to detect and prevent fraudulent activities such as item switching or under-scanning.

2.2.2 User Experience and Usability:

While self-checkout systems are convenient for many customers, others may find them confusing or challenging to use. Research can focus on improving user interfaces, accessibility, and the overall shopping experience to cater to a diverse range of customers.

2.2.3 AI and Computer Vision:

Advances in AI and computer vision technologies could enable more sophisticated self-checkout systems that can automatically recognize and process items. Research in this area can explore how to improve the accuracy and speed of these technologies.

2.2.4 Maintenance and Technical Issues:

Self-checkout systems can encounter technical issues such as software glitches or hardware malfunctions. Research can explore how to improve the reliability and robustness of these systems to minimize downtime and customer frustration.

2.3 Problem Formulation

In retail stores, it has been observed that there are longer queues for customers to checkout and pay the bills for the bought items. Though security cameras and high-level automated machines that scans the barcode are present in today's market, these are cost ineffective.

With the advent of technology, the retail sector has witnessed significant changes that once used to be believed too unfathomable. Self-service checkout systems have been established in a variety of stores in which customers have to scan all the bought items with a barcode reader and then pay for the bought items. However, these services could result in thefts in these stores and reduce customer experience and impersonal communication between employees and customers.

Technologies such as Amazon Go, released by Amazon in 2017, which is based on "Just Walk Out Technology", eased the self-checkout process as the customers do not have to wait in queues to pay for the purchased items. After installing the Amazon Go app, customers scan their iPhones as they enter the store. The track of their purchases is maintained. They can leave the store without physically checking out as their accounts are charged automatically. However, this technology can hinder customer experience as customers unfamiliar with the app would find it very difficult to shop. Besides the technology used in apps like Amazon Go is expensive.

Hence, there needs to be a system that is cost effective and can easily generate the bill without much human effort.

CHAPTER 3
PROPOSED SYSTEM

3.1 Proposed System

The proposed system takes image as input and generates a bill containing the price and quantity of the bought items.

The dataset contains 170-170 images of each of the products in various categories. These products include Apples, Oranges, Bananas and Eggs. We have used Robo Flow, which is a development framework for Computer Vision, used for enhancing data collection to pre-process the data and train the model. It is used to meet the following two objectives:

1) Pre-processing

There is an inbuilt feature in Roboflow for giving annotations. We had put labels on each of the photos. Apart from this, Roboflow also rotated the image from different angles. It was done for every image and hence 2000 approx. images were achieved in comparison to the earlier 640 images.

2) Data Production

In this process, Roboflow divided our dataset into three folders, that is, Training, Testing, and Validation. In each of these folders, there were 2 subfolders - images and labels. Data.yml files extracted all the labels that we gave and then allotted respective labels on respective items. After processing the dataset from Roboflow we received 1 API key and this API key was used to import data in Yolo for further detection.

We have used tkinter, which is a standard GUI library for Python, to provide the user interface. The users can upload the images or videos using this interface and the results get displayed.

YOLO is a well-recognized object detection method used widely because of its accuracy and quickness. We have used YOLOv5 which is a model trained on the COCO dataset which uses a transfer learning technique. We have tuned this model on our dataset to get better accuracy for object detection. YOLOv5 has five models: yolov5n, yolov5s, yolov5m, yolov5l and yolov5x. In our model, we have used yolov5s which has 7.2 M parameters.

YOLOv5 uses Efficient Det, which is a more complex architecture unlike the previous versions of YOLO. This helps to getter better accuracy and improved generalization of object categories. Apart from this, YOLOv5 is trained on 600 object categories which is a more diverse dataset than other versions. It also utilizes Spatial Pyramid Pooling (SPP) which helps to improve the performance of the model in detecting the smaller objects.

In Figure 3.1, we have explained the whole process of checkout at the retail stores with the help of the workflow diagram. When an input image containing the items is passed to the system, the system detects whether the data exists in the dataset or not. If yes, then the pre-trained model detects and recognizes the object. Then the detect method, which makes use of Tkinter, has a list that contains the per unit cost of each item. The bill is then created in the data frame and the price of each item is calculated. These price values are then added to the data frame and output is displayed.

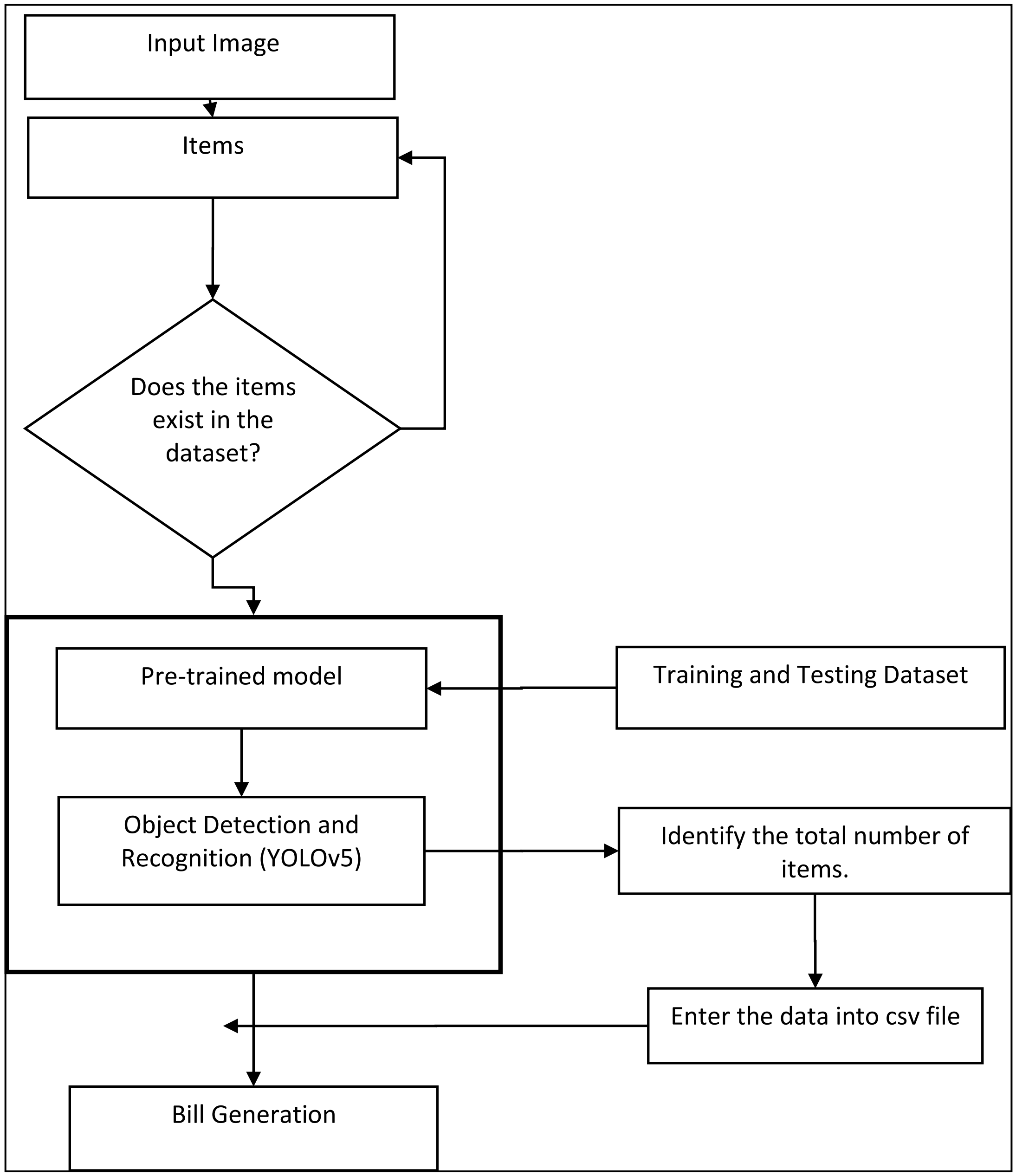


Fig. 3.1. Diagrammatic Representation of Proposed Methodology

3.2 Unique Features Of The System

Following are the unique features of the system that sets this system apart from the existing systems available in the market.

3.2.1 Customization and Training:

YOLOv5 allows users to train the model on their specific dataset. This is beneficial for adapting the system to the unique characteristics of a particular retail environment, including variations in lighting, item appearance, and camera angles.

3.2.2 Multi-class Object Detection:

YOLOv5 is capable of detecting multiple classes of objects simultaneously in a single frame. In retail checkout systems, this feature is crucial for identifying various products that may be present in the shopping basket.

3.2.3 User-friendly API:

Tkinter is used to provide a friendly interface to the users.

3.2.4 High Accuracy:

YOLOv5 is designed to provide high accuracy in object detection tasks.

3.2.5 Open-Source Community Support:

YOLOv5 benefits from an active open-source community, which means developers can access a wealth of resources, documentation, and community support when implementing and customizing the system.

CHAPTER 4
REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study

Conducting a feasibility study for an automated retail checkout system using YOLOv5 involves evaluating its technical, economic, and operational aspects.
Following are the various components:

4.1.1 Technical Feasibility:

4.1.1.1 Hardware Requirements:

Ultralytics YOLO can be run on a variety of hardware configurations, including CPUs, GPUs, and even some edge devices. However, for optimal performance and faster training and inference, a GPU with a minimum of 8GB of memory. NVIDIA GPUs with CUDA support are ideal for this purpose. This includes GPUs or CPUs capable of handling real-time object detection tasks, sufficient memory, and storage for model deployment and data processing.

For our system, we have used CUDA:0 device identifier and NVIDIA GPU (Tesla T4) for training YOLOv5 with the custom dataset. 2 CPUs core and 12.7 GB of RAM with a disc space of 166.8 GB were available.

4.1.1.2 Software Infrastructure:

The software stack required to deploy and integrate YOLOv5 into the retail checkout system includes frameworks such as TensorFlow, PyTorch, OpenCV, PyYAML and TorchVision. Apart from these, matplotlib, pandas, numpy, scipy are used to run the model. All these requirements are given in requirements.txt file.

Python 3.8 or later with all requirements dependencies installed, including torch>=1.7 comes in the txt file named as "requirements.txt". It contains all the libraries and their versions that are required in order to use YOLOv5 to train the custom dataset.

To install these dependencies, the dollowing command is run:

```
$ pip install -r requirements.txt
```

4.1.1.3 Data Collection and Annotation:

It is quite feasibile to collect and annotate the dataset for training the YOLOv5 model. The dataset is taken from Kaggle that contains images of Orange, Banana, and Apple. We have used Robo Flow, which is a development framework for Computer Vision, used for enhancing data collection to pre-process the data and train the model. It annotated the images and divided our dataset into 3 folders: Training, Validation and Testing. The time required for data collection and annotation is much less.

4.1.1.4 Model Training and Optimization:

The model is trained for 100 epochs with image size of 640 and batch size of 32. The confidence threshold for validation was of 0.25. It is quite feasible but running the model for 100 epochs required much time and patience.

The result accuracy can be improved by increasing the size of dataset and in order to reduce prediction accuracy, run the model on high end hardware.

4.1.2 Economic Feasibility:

4.1.2.1 Cost of Hardware and Software:

The costs associated with acquiring the necessary hardware components, including GPUs, CPUs, and storage devices is absolutely free as Google Colab and Kaggle notebooks with free GPU is provided.

Only the initial set up cost that includes a laptop or n laptops (in case of using it in multiple devices) is required. There is not much additional cost.

4.1.2.2 Development and Implementation Costs:

The costs associated with hiring personnel or outsourcing development tasks related to data collection, model training, and software integration is minimal. Most of the time was required to test and debug before deploying the system. Since manual testing was used to test the system, there were numerous test cases run but no additional software was required.

4.1.2.3 Operational Costs:

The ongoing operational costs associated with maintaining and supporting the automated retail checkout system is minimal. Most of the effort is done in updating the dataset in accordance to the available stock and updated pricing of the products. This includes personnel costs for system monitoring, maintenance, and updates. There is no additional costs such as electricity consumption or any recurring licensing or subscription fees.

4.1.3 Operational Feasibility:

4.1.3.1 User Acceptance and Training:

It is quite feasibile for customers. The training requirements for retail staff to operate and maintain the system effectively include not more than few hours and very few resources.

4.1.3.2 Scalability and Flexibility:

The system is scalable enough to handle multiple products by adding the products to the list. The system is flexible enough to adapt to changes in product offerings, retail layout, and customer preferences over time.

4.1.3.3 Risk Assessment:

The potential risks and challenges associated with the deployment and operation of the automated retail checkout system include hindrance in generating bill due to power cutoff or system shutdown due to technical problems.

4.2 Software Requirement Specification Document

4.2.1 Data Requirement

4.2.1.1 Data Sources

In order to train YOLOv5 on custom dataset, we have first imported the model from the below repository:

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
```

4.2.1.2 Data Set

The image dataset collected comprises of 50-50 images each of the products in various categories, in csv file format. These products include: - Orange, Banana, Apple and Egg.

We have used Robo Flow, which is a development framework for Computer Vision, used for enhancing data collection to pre-process the data and train the model.We have taken the dataset from Kaggle.

4.2.2 Functional Requirement

4.2.2.1 Object Detection

YOLOv5 is able to identify and categorize the objects efficiently. YOLOv5 has five models: yolov5n, yolov5s, yolov5m, yolov5l and yolov5x. In our model, we have used yolov5s which has 7.2 M parameters.

This helps to getter better accuracy and improved generalization of object categories. Apart from this, YOLOv5 is trained on 600 object categories which is a more diverse dataset than other versions.

4.2.2.2 Bill Generation

A bill that contains the total price of the bought products as well as their individual quanties and price, is generated. The bill is displayed using Tkinter which is a Python GUI that helps the users to interact with the system. The users can upload the images or videos using this interface and the results get displayed.

4.2.3 Performance Requirement

4.2.3.1 Speed and Responsiveness

The speed of generating the bill should be fast taking time less than 10s. YOLOv5 is fast in classifying the images. The expected speed and responsiveness of the system during different operations, such as scanning items is estimated to be much lesser than manual scanning of items.

4.2.3.2 Object Detection Accuracy

The confidence value for object classification should be atleast 70%. A confidence score is calculated as an evaluation standard. This confidence score shows the probability of the image being detected correctly by the algorithm and is given as a percentage. The scores are taken on the mean average precision at different IoU (Intersection over Union) thresholds.

4.2.4 Maintainability Requirement

4.2.4.1 Code Maintainability

All the documents such as Software Requirement Specification Document, Test Plan, Project Synopsis and Project Report is maintained. The code is maintained on github.

4.2.5 Security Requirement

4.2.5.1 Access Control

The access of the system should be controlled by the retailers and only they can decide which employee the access should be given. This helps to control the security of the store and only the authorized ones can get the access to it

4.3 SDLC Model

Software Development Life Cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software. SDLC, or software development life cycle, is a methodology that defines the entire procedure of software development step-by-step. A software development life cycle (SDLC) consists of 6 stages: Planning, Analysis, Design, Implementation, Testing and Maintainence.

The test methodology selected for this project is Agile. An Agile methodology is the most suitable for this project. It allows for flexibility, ongoing testing, and adaptation, which are essential for projects that involve machine learning, image processing, and AI.

Agile enables us to respond to changing requirements and refine the style transfer algorithm as you gain insights from testing and user feedback. It is sequential development process that flows like a waterfall through all phases of a project (analysis, design, development, and testing, for example), with each phase completely wrapping up before the next phase begins. The figure 4.3 demonstrates the stages of agile methodology.

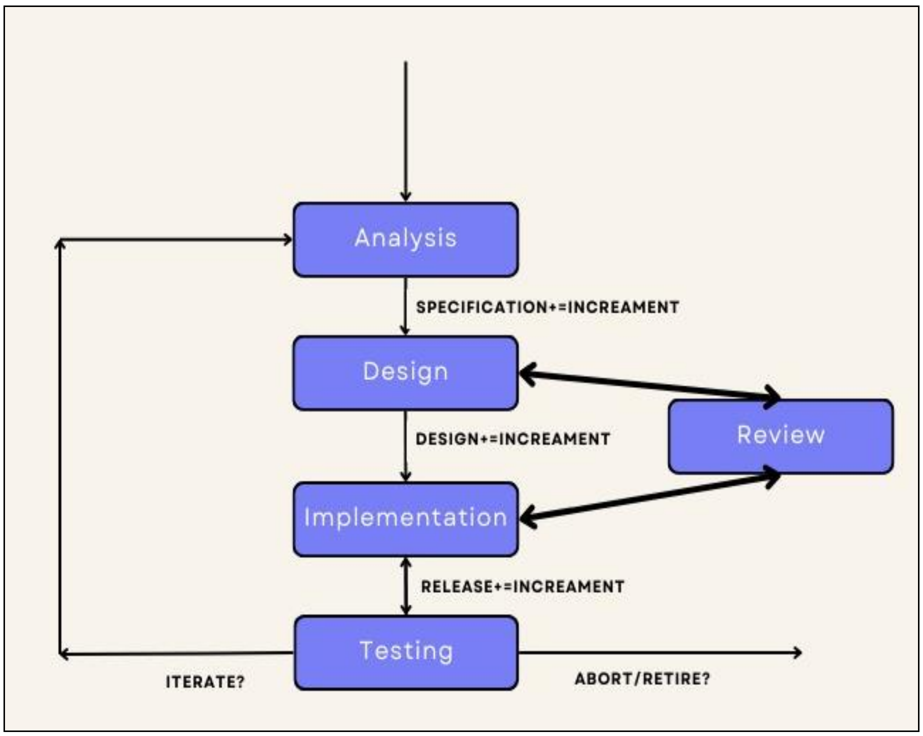


Fig. 4.1. Stages of Agile Methodology [16]

The phases of Agile are Analysis, Design, Implementation and Testing. A brief description of these phases as shown in the figure above is detailed as:

4.3.1 Stage 1: Analysis

4.3.1.1 Product Backlog Creation:

It involved discussing and gathering requirements such as features of the system, methodologies to be used, outcome to be produced, testing criteria, etc.

4.3.1.2 Identifying MVP (Minimum Viable Product):

The minimum vialble product includes the core functionalities required for the initial release. In our system, the MVPs were item recognition and receipt generation.

4.3.2 Stage 2: Design

4.3.2.1 Sprint Planning:

The backlog items were broken into smaller tasks for implementation. They were: Project Setup and Data Collection, YOLOv5 Model Training, User Interface, Testing and Bug Fixing, and Documentation.

Establishing sprint goals, assigning tasks to team members and planning sprint time line came under sprint planning.

4.3.2.2 Architecture Design:

The overall system architecture including the Data Flow Diagrams (Level 0 and Level 1), Use Case Diagrams, Entity-Relationship Diagram and user interface design for the checkout process, was done at this stage.

A workflow diagram was also generated to get a better understanding of the system processes. It demonstrated the complete flow of all the activities taking place.

4.3.3 Stage 3: Implementation

4.3.3.1 Development Sprints:

All the features mentioned in sprint planning stage were done in short iterations. The timeline was to complete the sprints in 2-3 weeks.

4.3.3.2 Continuous Integration:

It involved integrating YOLOv5 model into the system and ensure it detects items accurately. Single class object detection and multi class object detection was done to check the accuracy of the model.

4.3.3.3 Unit Testing:

It included writing and executing unit tests to verify the functionality of individual components. It helped to find the errors in the respective modules and helped to solve these errors.

4.3.3.4 Pair Programming:

Through pair programming, we aimed to encourage collaboration among the team members to share knowledge and to ensure code quality.

4.3.4 Stage 4: Review

4.3.4.1 Sprint Review:

At this stage, we shared the completed features to our project guide and peers and gather feedback for further improvements. The feedback taken was then incorporated into the system to improve the functionality of the system.

4.3.5 Stage 5: Testing

There are three types of testing that we have used in this system:

4.3.5.1 White Box Testing:

It involved examining the internal structure and logic of the software application.

4.3.5.2 Regression Testing:

It ensured that new changes do not introduce bugs or regressions in existing functionalities. It was done to test the system’s compatibility of integrating the new changes in existing functionalities and make sure that the system is free from any errors.

4.3.5.3 Black Box Testing:

It involving testing the system's functionalities without peering into its internal structure or working .

4.3.6 Stage 6: Deployment and Maintenance

4.3.6.1 Release Planning:

The release schedule was planned based on feedback from testing phases.

4.3.6.1 Bug Fixes:

It involved addressing any bugs or issues reported by the peers while reviewing the system and fixing them as soon as possible.

4.3.6.2 Feature Enhancements:

It included improving the system continuously based on user feedback and changing requirements. The changing requirements mean that the products to be detected are added to the datset and the model is trained accordingly. The prices of the new products need to be added to the list as well.

Throughout the entire process, collaboration, adaptability, and customer satisfaction should remain central principles.

4.4 System Design

4.4.1 Data Flow Diagrams

4.4.1.1 DFD Level 0

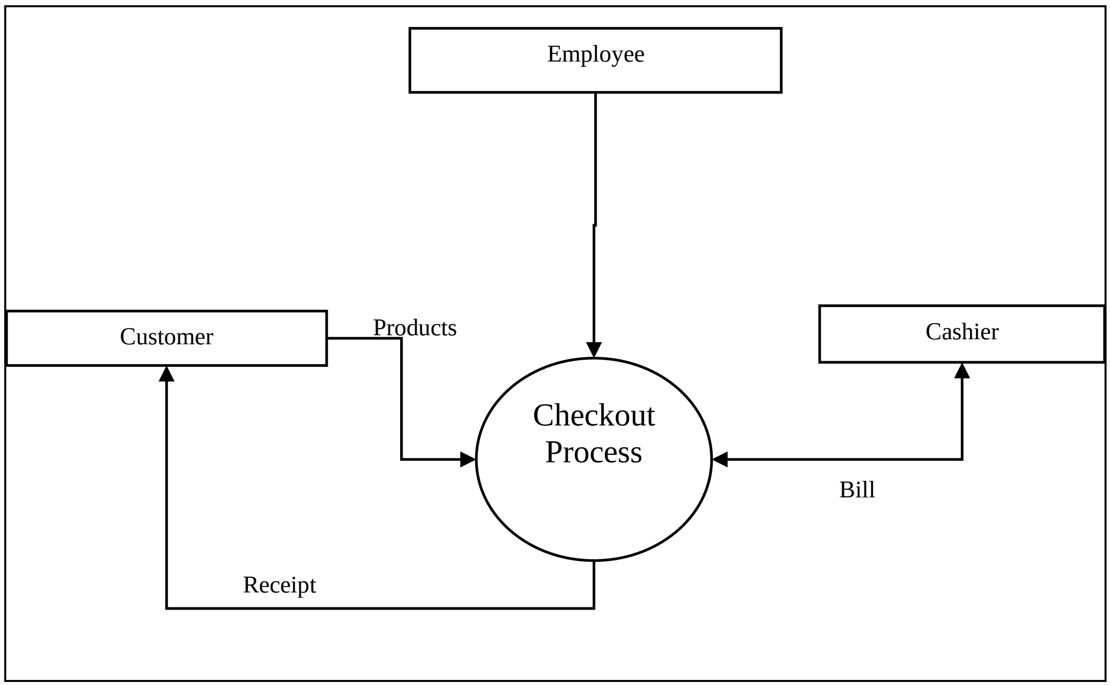


Fig. 4.2. 0 Level DFD

The above figure shows the 0 Level Data Flow Diagram (DFD) of Automated Retail Checkout System using YOLOv5 in Computer Vision.

This is the highest-level DFD, which provides an overview of the entire system. It shows the major processes, data flows, and data stores in the system, without providing any details about the internal workings of these processes. The major process is Checkout Process and the external entities are Customer, Cashier and Employee.

4.4.1.2 DFD Level 1

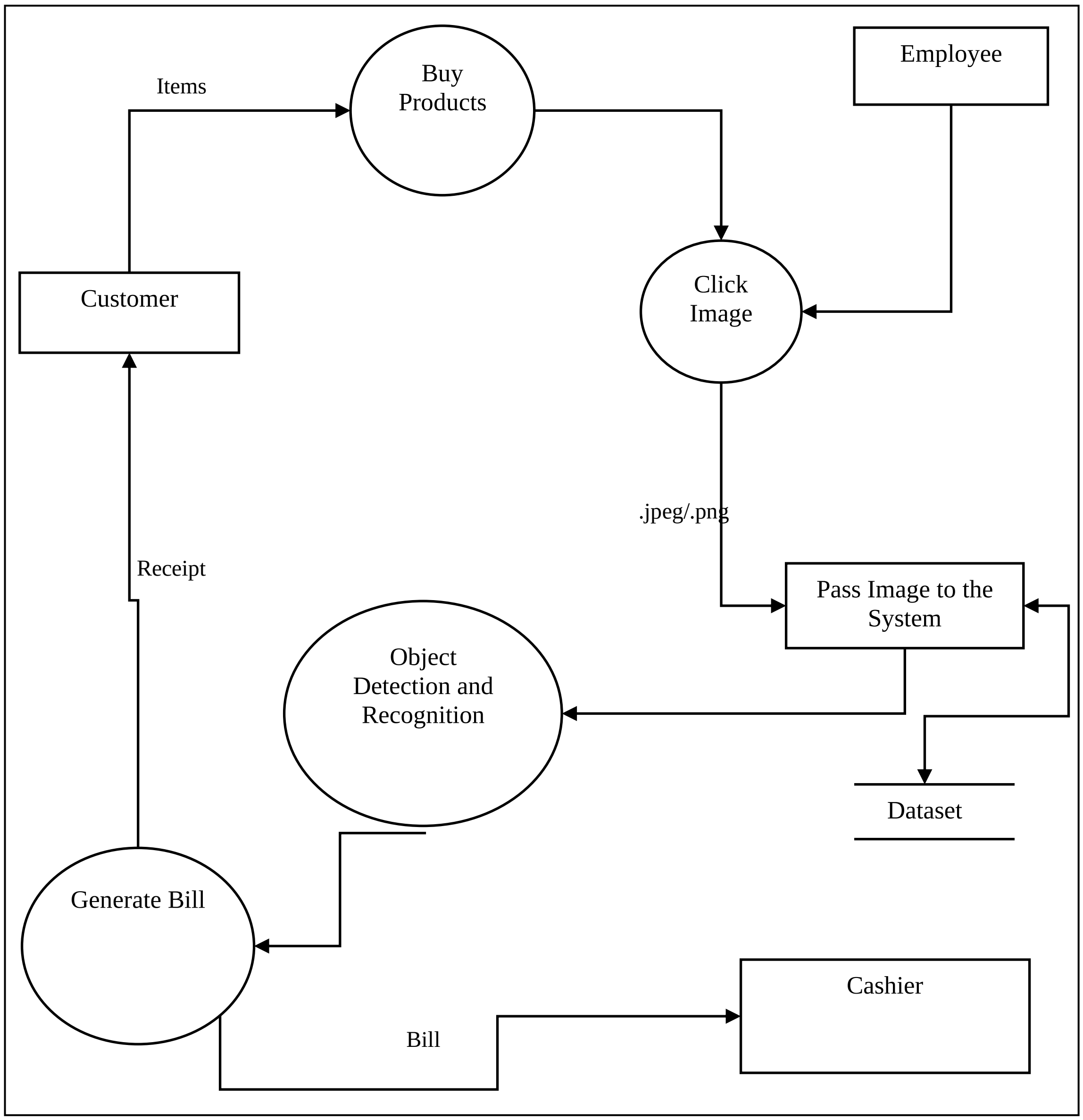


Fig. 4.3. 1 Level DFD

The above figure shows the 1 Level Data Flow Diagram (DFD). This is the highest-level DFD provides a more detailed view of the system.

4.4.2 Use Case Diagram

Fig. 4.4. Use Case Diagram

The main components of the use case diagram are the actors and the use cases. The actors are customer, manager and cashier. Customer is the person who uses the automated checkout systemn, manager is responsible for managing the system, updating the product database, and handling exceptions, and cashier carries out the checkout process. The use cases are click image, pass image to the system, display bill, payment and view products.

4.5 Database Design - ER Diagram

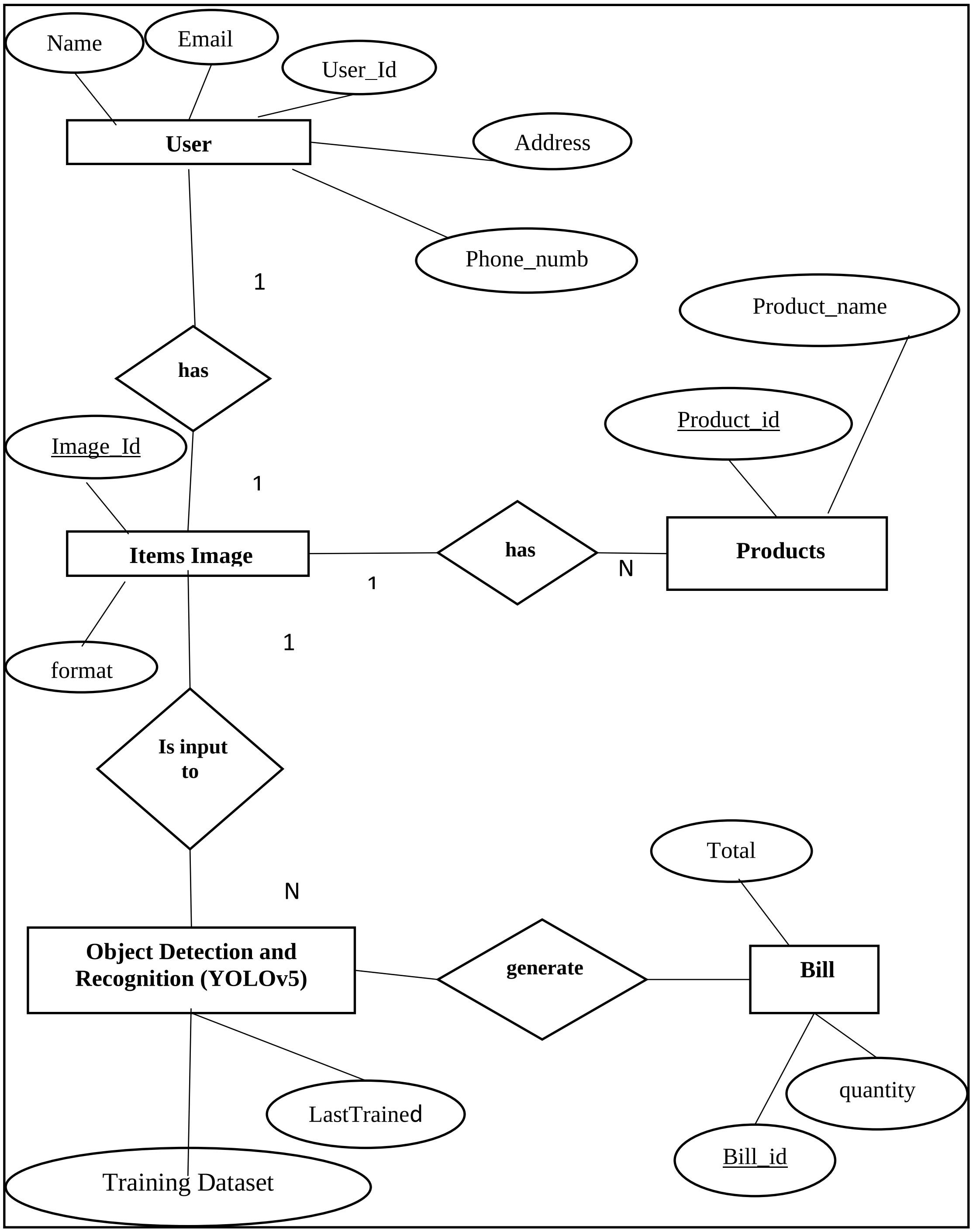


Fig. 4.5. ER Diagram

CHAPTER 5
IMPLEMENTATION

5.1 Introduction to Languages, Tools And Technologies Used for Implementation

5.1.1 Language used:

Python: The whole code is written in Python programming language.

5.1.2 Toolkit:

Roboflow: It is a Computer Vision developer framework used for better data collection, pre-processing, and model training techniques.

YOLOv5: It is a novel convolutional neural network (CNN) that detects objects in real-time with great accuracy. This approach uses a single neural network to process the entire picture, then separates it into parts and predicts bounding boxes and probabilities for each component.

5.1.3 User Interface:

Tkinter: Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications.

5.1.4 Coding Environment:

Visual Studio Code: Visual Studio Code is a source-code editor.

Google Colab: It is a hosted Jupyter Notebook service that requires no setup to use and provides free access to computing resources, including GPUs and TPUs. It is used to train the Model and to write the code for the Tkinter interface.

CHAPTER 6

TESTING AND MAINTENANCE

6.1 Testing Techniques And Test Cases Used

In automated retail checkout system using YOLOv5, we have used the technique of manual testing. Manual testing is a technique to test the software that is carried out using the functions and features of an application. There are different methods to implement manual testing, but it is broadly classified into three types of manual testing: White Box Testing, Black Box Testing and Gray Box Testing.

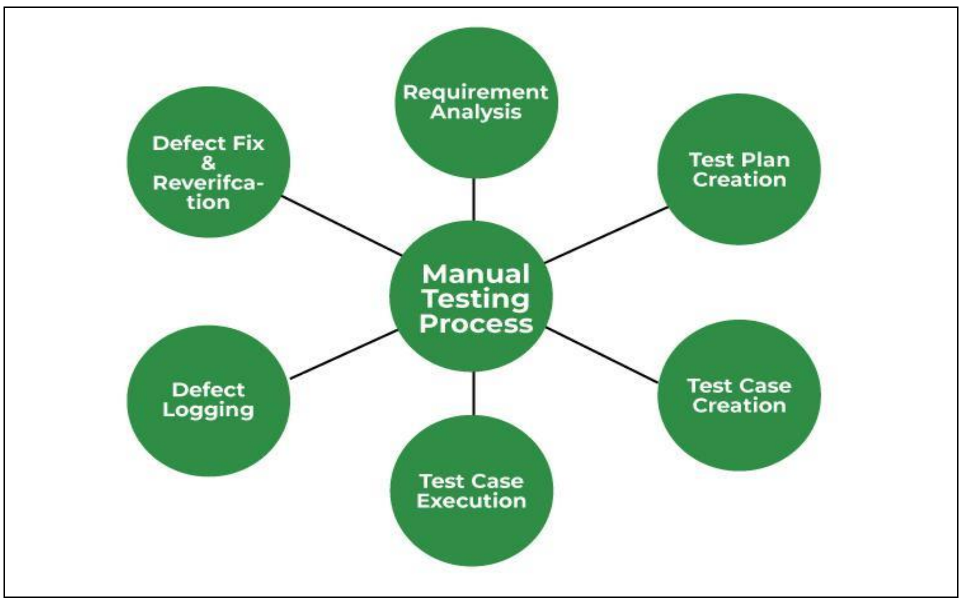


Fig. 6.1. Process of Manual Testing [17]

Both white-box and black-box testing techniques are employed to ensure the quality and reliability of the system. By combining white-box and black-box testing techniques, automated retail checkout system can undergo comprehensive testing to validate its functionality, reliability, usability, and security. This approach helped us to identify and address defects at various levels of the software development lifecycle, ensuring the delivery of a robust and high-quality system to end-users.

6.1.1 White-Box Testing:

White-box testing, also known as structural or glass-box testing, involves examining the internal structure and logic of the software application. It is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality. In white-box testing, an internal perspective of the system is used to design test cases. The process is entirely carried out manually and the process is efficient since the checking code or design is manually checked by humans.

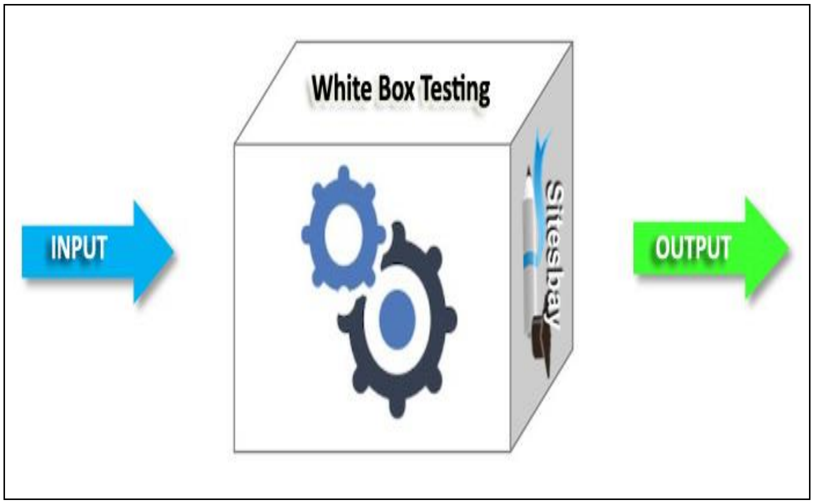


Fig. 6.2. White Box Testing Approach [17]

In the context of our project, white-box testing techniques includes the following testing techniques:

6.1.1.1 Code Coverage Analysis:

It ensured that all lines of code, branches, and statements were executed during testing. Techniques such as statement coverage, branch coverage, and path coverage could be used to analyze code coverage.

6.1.1.2 Unit Testing:

It aimed to test individual components (units) of the system, such as functions, methods, or modules. It verified that each unit behaved as expected and produced the correct output for different inputs.

6.1.1.3 Integration Testing:

It involed testing the interaction and integration between different modules or components of the system. It verified that modules worked together seamlessly and exchange data correctly.

6.1.1.4 Code Review:

The team members along with few batchmates conducted peer reviews and code inspections to identify defects, logic errors, or coding standards violations. It helped to review the source code to ensure it is well-structured, readable, and maintainable.

6.1.2 Black-Box Testing:

Black-box testing, also known as functional or specification-based testing, focuses on testing the external behavior of the system without considering its internal implementation details. It is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. This method of test can be applied virtually to every level of software testing: unit, integration, system and acceptance.

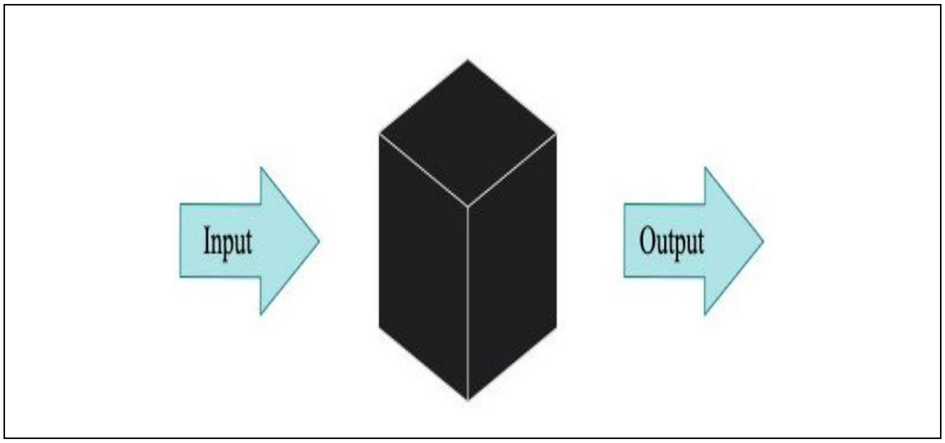


Fig. 6.3. Black Box Testing Approach [17]

These techniqus are described as mentioned below:

6.1.2.1 Functional Testing:

It tested the system's functionality against the specified requirements and user expectations.

We executed test cases to verify that the system performs the intended operations accurately and produces the expected results.

6.1.2.2 User Interface Testing:

It involved testing the graphical user interface (GUI) of the system to ensure usability, consistency, and responsiveness. It verified that users can interact with the system effectively and intuitively.

6.1.2.3 Regression Testing:

We have re-run previously executed test cases to ensure that recent changes or enhancements have not introduced new defects or regression issues. It helped to verify that existing functionality remains intact after system updates or modifications.

6.1.2.4 Usability Testing:

It evaluated the system's ease of use, accessibility, and user experience from the perspective of end-users. It helped to tdentify any usability issues, navigation problems, or ambiguities in the system's workflow.

6.1.2.5 Security Testing:

It assessed the system's security posture by testing for vulnerabilities, such as unauthorized access, data breaches, or injection attacks.

6.1.3 Test Cases Used:

6.1.3.1 Object Detection:

It verifies that the system accurately detects and identifies the items in the image.

6.1.3.2 User Interface:

It tests the user interface to ensure a smooth and intuitive checkout process. It also checks whether the user interface takes image input or not.

6.1.3.3. Performance:

It tests the system's performance under normal operating conditions to ensure real-time object detection and checkout processing.

6.1.3.4 Correct Bill Generation:

It verifies that the model calulates the bill correctly and there are no errors in detecting the price and quantity.

6.1.3.5 Image Format:

The image format that the system takes as input must be either .jpeg or .png. It must successfully run on these image formats.

The table below contains the various test cases that were used to test the system for its functionality and performance.

Table 6.1 Description of Test Cases Used

Test Case No.	**Test Case Name**	**Test Case Description**
1.	Detect Orange	The system must be able to detect and classify orange.
2.	Detect Apple	The system must be able to detect and classify apple.
3.	Image Format - .jpeg	The system must be able to take .jpeg input image.
4.	Detect multiple items	The system must be able to detect multi class items.
5.	Display total price – single class	The system must be able to detect and classify objects belonging to same class and generate a bill.
6.	Display total price – multiple class	The system must be able to detect and classify objects that belong to different classes and generate a bill.
7.	Image Format - .png	The system must be able to tale .png as input image.

In the table 6.2, the data is recorded that shows the bug report containing the input, expected output, actual output and result of running the test cases.

Table 6.2 Bug Report

Test Case No.	**Test Case Name**	**Input**	**Expected Output**	**Actual Output**	**Test Result – Pass/Fail**
1.	Detect Orange	Orange Image	Orange is predicted	Orange is predicted	Pass
2.	Detect Apple	Apple Image	Apple is predicted	Apple is predicted	Pass
3.	Image Format - .png	Orange.jpeg image	Accepted	Accepted	Pass
4.	Detect multiple items	Image – apple, banana, orange	1 apple, 1 banana, 1 orange is predicted.	1 apple, 1 banana, 1 orange is predicted.	Pass
5.	Display total price – single class	3 apples	290	290	Pass
6.	Display total price – multiple class	2 apples, 2 bananas, 2 oranges	200	200	Pass
7.	Image Format - .jpeg	Apple.jpeg image	Accepted	Accepted	Pass

CHAPTER 7
RESULTS AND DISCUSSIONS

7.1 Presentation of Results

7.1.1 Model summary on training dataset

After training the YOLOv5 model on the custom dataset, we get the following results:

YOLOv5s summary: 213 layers, 7023610 parameters, 0 gradients, 15.8 GFLOPs

Table 7.1 Model Summary on Training Dataset

Class	**Images**	**Labels**	**P**	**R**	**mAP@.5**	**mAP@.5:0.95**
All	640	880	0.945	0.928	0.955	0.575
Orange	170	220	0.988	1	0.995	0.602
Banana	170	220	0.868	1	0.957	0.563
Apple	170	220	0.978	1	0.995	0.617
Egg	170	220	0.904	0.783	0.845	0.453

Precision of 0.945 and Recall of 0.928 are both high, indicating that the model is good at identifying true positives and minimizing false positives.

mAP@.5 (mean Average Precision with 0.5 IoU) of 0.955 is excellent, showing that the model performs well with standard object detection tasks.

mAP@.5:.95 (with varying IoU thresholds) of 0.575 indicates that while the model is good at detecting objects with a relatively lower IoU threshold, it has more challenges with stricter thresholds.

Orange, Banana and Apple have high precision and recall, with mAP@.5 values close to or above 0.98, suggesting excellent detection for these classes.

7.1.2 Visual Representation of Model Summary on Training Dataset

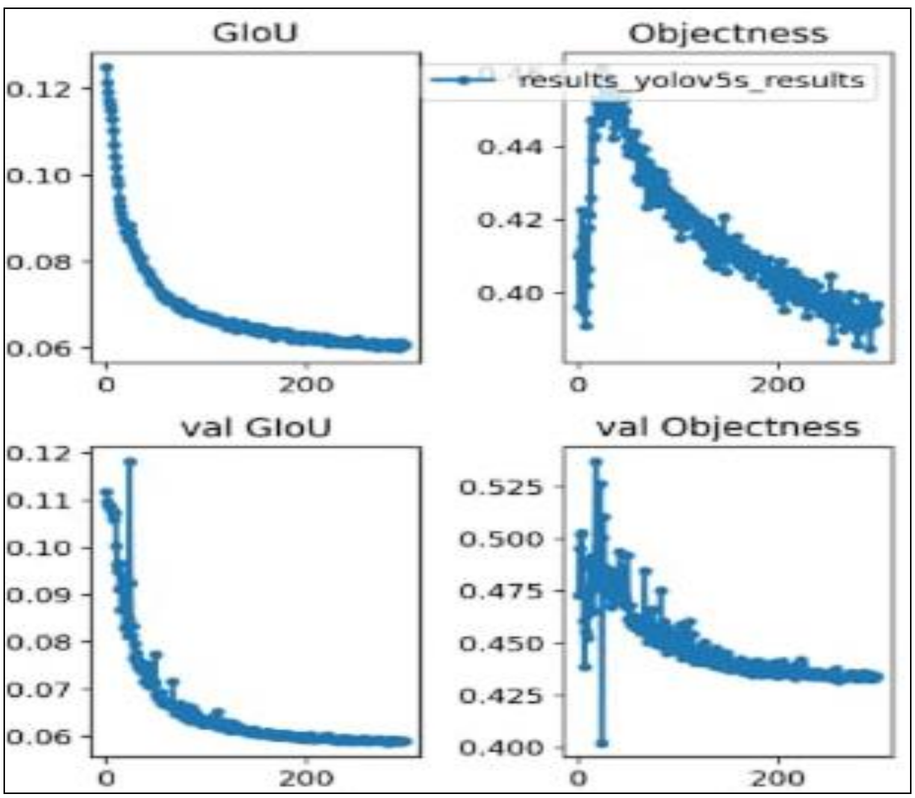


Fig. 7.1. Visual Representation of Model Summary on Training Dataset

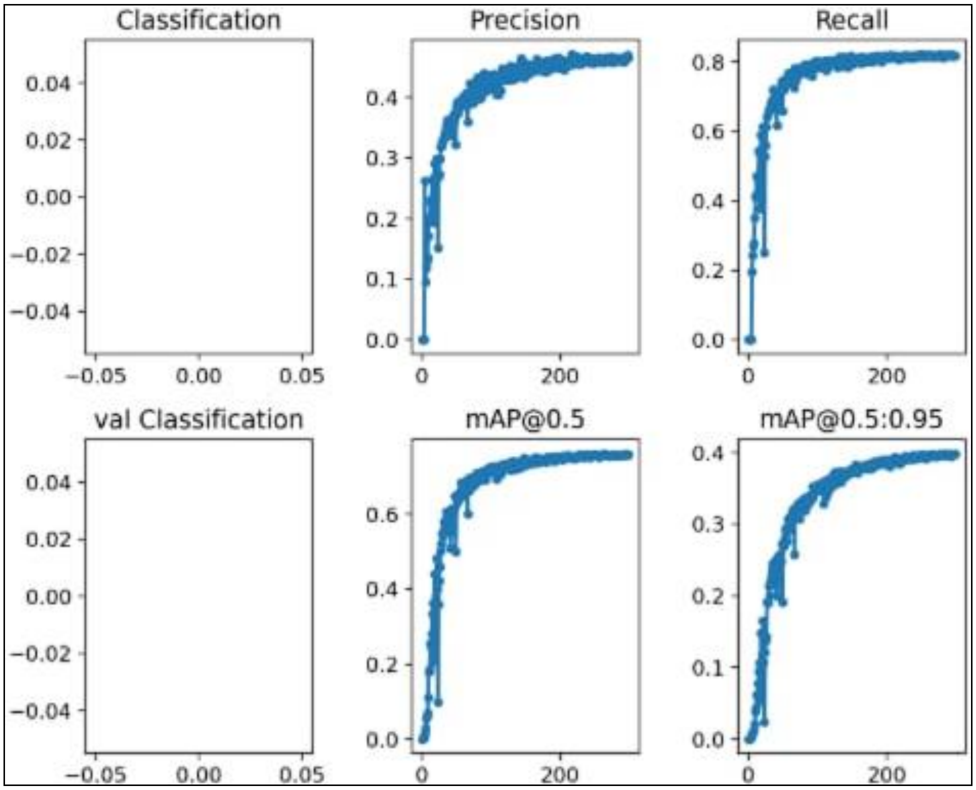


Fig. 7.2. Visual Representation of Model Summary on Training Dataset

7.1.2.1 GIoU (Generalized Intersection over Union):

This metric evaluates the overlap between the predicted bounding boxes and the ground truth boxes.

The plot shows the GIoU loss decreasing over iterations, indicating that the model is getting better at predicting bounding boxes that closely match the ground truth.

7.1.2.2 Objectness:

This measures the confidence that an object exists in the predicted bounding boxes. The plot shows the objectness score during training. A high score means the model is confident in its object predictions. The gradual decrease and stabilization suggest that the model is becoming more confident in its predictions over time.

7.1.2.3 Classification:

This metric tracks the classification loss for the detected objects.

The plot shows the classification loss over iterations. The values moving closer to zero indicate that the model is improving in correctly classifying the detected objects.

7.1.2.4 Precision:

Precision is the ratio of correctly predicted positive observations to the total predicted positives.

The plot shows an increase in precision over time, indicating that the model is becoming better at correctly identifying objects and minimizing false positives.

7.1.2.5 Recall:

Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

The plot shows an increase in recall, suggesting that the model is becoming better at detecting all relevant objects and minimizing false negatives.

7.1.2.6 Validation GIoU:

Similar to the GIoU in the training phase, but evaluated on the validation dataset.

The plot shows a decreasing trend, indicating that the model's bounding box predictions are improving on the validation set as well.

7.1.2.7 Validation Objectness:

Similar to the objectness score in the training phase, but evaluated on the validation dataset.

The plot shows the validation objectness score, with a trend similar to the training objectness score, suggesting that the model's confidence in object predictions is consistent between training and validation datasets.

7.1.2.8 Validation Classification:

Similar to the classification loss in the training phase, but evaluated on the validation dataset.

The plot shows the validation classification loss. A decreasing trend would indicate that the model is also improving in classifying objects correctly on the validation set.

7.1.2.9 mAP@0.5 (Mean Average Precision at IoU threshold 0.5):

This metric evaluates the overall performance of the object detector at an IoU threshold of 0.5.

The plot shows an increasing mAP@0.5, indicating that the model's accuracy in detecting objects is improving over time.

7.1.2.10 mAP@0.5:0.95:

This metric evaluates the mean average precision across multiple IoU thresholds (from 0.5 to 0.95).

The plot shows an increasing trend, suggesting that the model's performance is improving across different levels of overlap between predicted and ground truth bounding boxes.

7.2 Performance Evaluation

7.2.1 Performance Metric Analysis

```
Run summary:
          best/epoch 94
        best/mAP_0.5 0.95466
   best/mAP_0.5:0.95 0.57594
      best/precision 0.94474
         best/recall 0.92808
     metrics/mAP_0.5 0.95464
metrics/mAP_0.5:0.95 0.57517
   metrics/precision 0.94485
      metrics/recall 0.92803
      train/box_loss 0.02262
      train/cls_loss 0.00266
      train/obj_loss 0.01043
        val/box_loss 0.0164
        val/cls_loss 0.00079
        val/obj_loss 0.00376
               x/lr0 0.0003
               x/lr1 0.0003
               x/lr2 0.0003
```

Fig. 7.3. Performance Metrics

7.2.1.1 Best Epoch:

It indicates the epoch with the best results.

7.2.1.2 mAP (mean Average Precision) 0.5:

This measures the precision of the model with a 0.5 IoU (Intersection over Union) threshold. A high mAP of 0.95466 is quite impressive.

7.2.1.3 mAP 0.5:0.95:

This is a more stringent metric, averaging the mAP over IoU thresholds from 0.5 to 0.95, indicating generalization. The value of 0.57594 suggests there might be room for improvement in dealing with varied object sizes or positions.

7.2.1.4 Precision and Recall:

Precision of 0.94474 and Recall of 0.92808 indicates that the model performs well in terms of reducing false positives and maximizing correct detections.

7.2.1.5 Validation Losses:

Box Loss: It measures how well the predicted bounding boxes match the ground truth. A lower value (0.0164) is good.

Class Loss: It reflects how well the model classifies the detected objects. With 0.00079, this seems quite low, indicating accurate classification.

Object Loss: This loss evaluates objectness, indicating if the model effectively distinguishes between objects and background. A lower value of 0.00376 is positive.

7.2.1.6 Learning Rates:

The three learning rates (lr0, lr1, lr2) are consistent at 0.0003, indicating the same rate for all layers. This suggests a stable learning process.

7.2.2 Detection on Test Samples

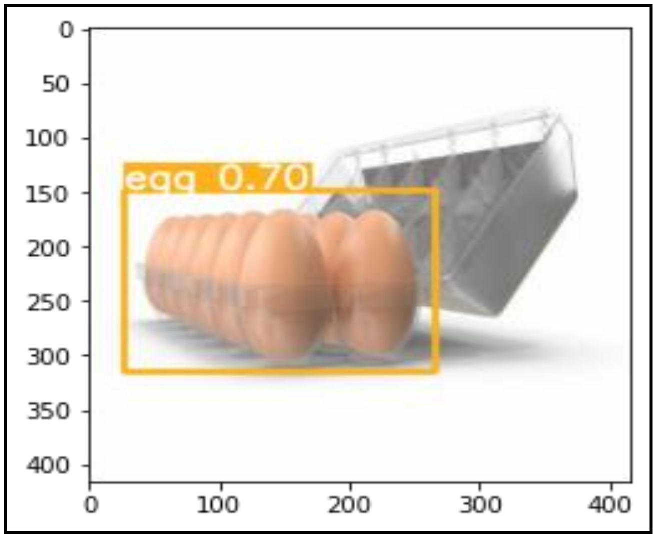


Fig. 7.4. Prediction Value on Test Sample (Egg)

Here we can see that the model works well in detection and egg is predicted with 70% confidence.

7.2.3 Confusion Matrix:

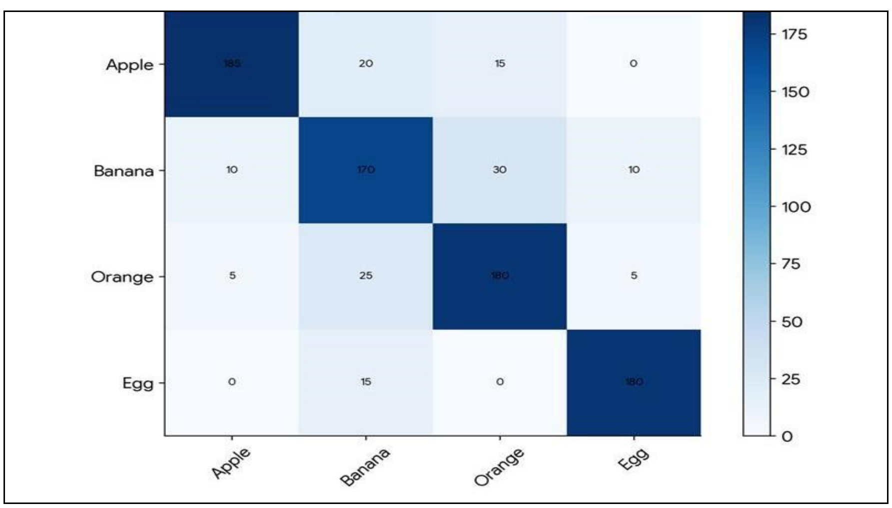


Fig. 7.5. YOLOv5 Object Detection Confusion Matrix

The confusion matrix is a visual representation used to evaluate the performance of an object detection model. It helps to understand how well the model is distinguishing between different classes. Each cell in the matrix represents the count of true positives, false positives, false negatives, and true negatives for each class.

7.2.3.1 Key Elements of the Confusion Matrix:

True Positives (TP): The number of correct predictions for a particular class. For example, the value at the intersection of the "Apple" row and "Apple" column (185) indicates that the model correctly predicted 185 apples.

False Positives (FP): The number of incorrect predictions where a class is incorrectly predicted as another class. For example, the value at the intersection of the "Apple" row and "Banana" column (20) indicates that 20 apples were incorrectly predicted as bananas.

False Negatives (FN): The number of incorrect predictions where a class is missed and predicted as another class. For example, the value at the intersection of the "Banana" row and "Apple" column (10) indicates that 10 bananas were incorrectly predicted as apples.

True Negatives (TN): Not explicitly shown in the confusion matrix but can be inferred from the matrix. These are the instances where the model correctly identified that an object does not belong to a specific class.

7.2.3.2 Interpreting the Given Confusion Matrix:

Apple:

True Positives: 185 (correctly predicted as apple)

False Positives: 10 (bananas incorrectly predicted as apple), 5 (oranges incorrectly predicted as apple)

False Negatives: 20 (apples incorrectly predicted as banana), 15 (apples incorrectly predicted as orange)

Banana:

True Positives: 170 (correctly predicted as banana)

False Positives: 20 (apples incorrectly predicted as banana), 25 (oranges incorrectly predicted as banana), 15 (eggs incorrectly predicted as banana)

False Negatives: 10 (bananas incorrectly predicted as apple), 30 (bananas incorrectly predicted as orange), 10 (bananas incorrectly predicted as egg)

Orange:

True Positives: 180 (correctly predicted as orange)

False Positives: 15 (apples incorrectly predicted as orange), 30 (bananas incorrectly predicted as orange)

False Negatives: 5 (oranges incorrectly predicted as apple), 25 (oranges incorrectly predicted as banana), 5 (oranges incorrectly predicted as egg)

Egg:

True Positives: 180 (correctly predicted as egg)

False Positives: None

False Negatives: 10 (bananas incorrectly predicted as egg), 5 (oranges incorrectly predicted as egg)

7.4 Key Findings

The system works well detecting real images as shown in the below figure.

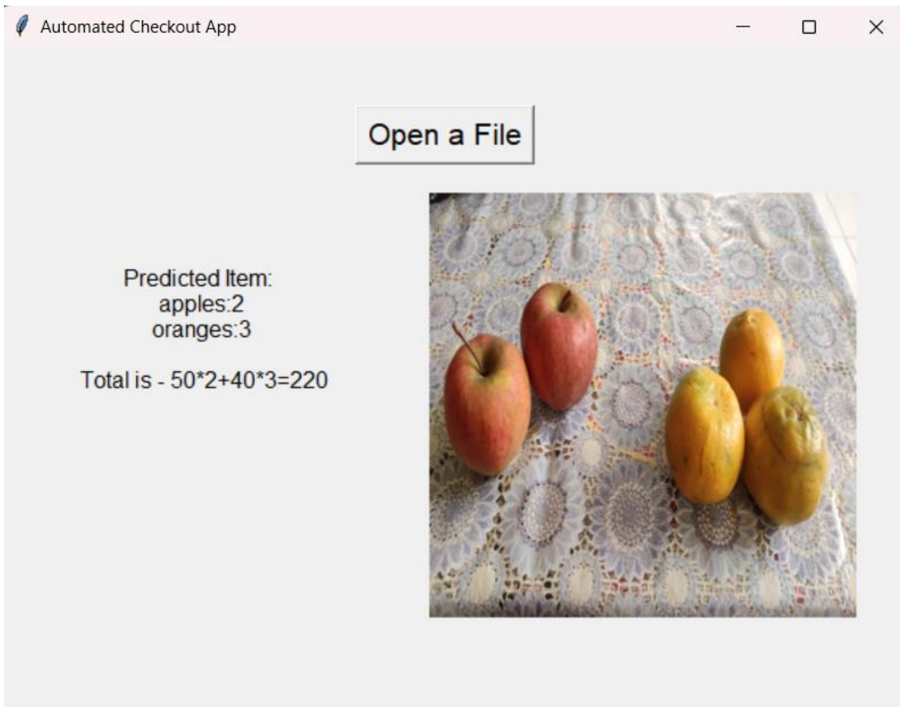


Fig. 7.6. Final Output Screen

In fig. 10, the input image contains 2 apples, and 3 oranges. The price of each quantity as mentioned in the list is ['banana': 10, 'apple':50, 'orange':40].

The predicted items were ['apples':2 , 'orange': 3]. Hence the bill generated is calculated by multiplying the quantity with price and then adding it to the final value. Therefore the output is 220.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

This system provides an innovative solution that can help the masses to smoothen their checkout experience. It can help the retailers and the customers to effectively checkout using a simple approach. It can generate the bill of the items that contain the information regarding the bought products like name of the time, quantity, weight, price as well as the total bill. The system uses YOLOv5, a very precise and efficient object detection model, to efficiently detect the objects in the video or the image. Since this model uses fewer resources and is cost-effective in comparison to the technologies existing in the market, it can provide an easier approach to checkout services. Retailers as well as customers can benefit from it. It provides a solution to the long queues the customers must stand in to pay the bill.

This system shows great promise but there are certain limitations and areas for further improvement. It is crucial for the retailers to maintain up-to-date datasets for object detection. Apart from this, there might be some errors in accurately identifying and classifying the objects in situations like poor lighting in photos or low-resolution pictures. The future work can be done regarding the improved and novel models for object detection tasks and enhancing the training strategies like transfer learning or domain learning.

Future scope is likely to continue evolving with advancements in computer vision and artificial intelligence. Integration with other emerging technologies such as RFID (Radio-Frequency Identification), IoT (Internet of Things), and edge computing can enhance the capabilities of automated retail checkout systems. This integration could enable more seamless and accurate tracking of products. Apart from this, future developments may focus on improving the user experience by reducing false positives, handling complex scenarios (e.g., crowded stores).

REFERENCES

[1] D. Triphena Delight and V. Karunakaran, “A Comprehensive Analysis of Methodologies used for Object Detection and Localization,” in 2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, Institute of Electrical and Electronics Engineers Inc., Mar. 2021, pp. 448–453.

[2] Y. Hu, X. Li, N. Zhou, L. Yang, L. Peng, and S. Xiao, “A Sample Update-Based Convolutional Neural Network Framework for Object Detection in Large-Area Remote Sensing Images,” IEEE Geosci. Remote Sens. Lett., vol. 16, no. 6, pp. 947–951, Jun. 2019,
doi: 10.1109/LGRS.2018.2889247.

[3] A. John and D. Meva, “A Comparative Study of Various Object Detection Algorithms and Performance Analysis,” Int. J. Comput. Sci. Eng. Open Access Res. Pap., vol. 8, no. 10, 2020, doi: 10.26438/ijcse/v8i10.158163.

[4] Shandong da xue, Asia-Pacific Signal and Information Processing Association, and Institute of Electrical and Electronics Engineers, 2019 2nd IEEE International Conference on Information Communication and Signal Processing (ICICSP 2019) : September 28-30, 2019, Weihai, China.

[5] D. A. Mora Hernandez, O. Nalbach, and D. Werth, “How computer vision provides physical retail with a better view on customers,” in

Proceedings - 21st IEEE Conference on Business Informatics, CBI 2019, Institute of Electrical and Electronics Engineers Inc., Jul.
2019, pp. 462–471. doi: 10.1109/CBI.2019.00060.

[6] N. Shekokar, A. Kasat, S. Jain, P. Naringrekar, and M. Shah, "Shop and go: An innovative approach towards shopping using deep learning and computer vision," in Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 1201–1206. doi: 10.1109/ICSSIT48917.2020.9214256.

[7] M. A. Majdi, B. Sena Bayu Dewantara, and M. M. Bachtiar, "Product stock management using computer vision," in IES 2020 -International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort, Institute of Electrical and Electronics Engineers Inc., Sep. 2020, pp. 424–429. doi: 10.1109/IES50839.2020.9231673.

[8] B. Dumnić et al., IEEE EUROCON 2019 : 18th International Conference on Smart Technologies: 1st - 4th July 2019, Novi Sad, Serbia.

[9] S. K. Yedla, V. M. Manikandan, and V. Panchami, "Real-Time Scene Change Detection with Object Detection for Automated Stock Verification," in ICDCS 2020 - 2020 5th International Conference on Devices, Circuits and Systems, Institute of Electrical and Electronics Engineers Inc., Mar. 2020, pp. 157–161. doi: 10.1109/ICDCS48716.2020.243571.

[10] M. Sugadev, K. Sucharitha, I. R. Sheeba, and B. Velan, "Computer vision based automated billing system for fruit stores," in Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Institute of Electrical and Electronics Engineers Inc., Aug. 2020, pp. 1337–1342. doi: 10.1109/ICSSIT48917.2020.9214101.

[11] H. Y. Putra, "Fraud detection at self checkout retail using data mining," in 2020 International Conference on Information Technology Systems and Innovation, ICITSI 2020 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Oct. 2020, pp. 211–216. doi: 10.1109/ICITSI50517.2020.9264919.

[12] "What is YOLOv5? A Guide for Beginners." Accessed: Oct. 31, 2023. [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/

[13] "YOLOv5 - Fine Tuning & Custom Object Detection Training."Accessed: Oct. 31, 2023. [Online]. Available: https://learnopencv.com/custom-object-detection-training-using-yolov5.

[14] "Object Detection: Models, Architectures & Tutorial [2023]."Accessed: Oct. 31, 2023. [Online]. Available: https://www.v7labs.com/blog/object-detection-guide

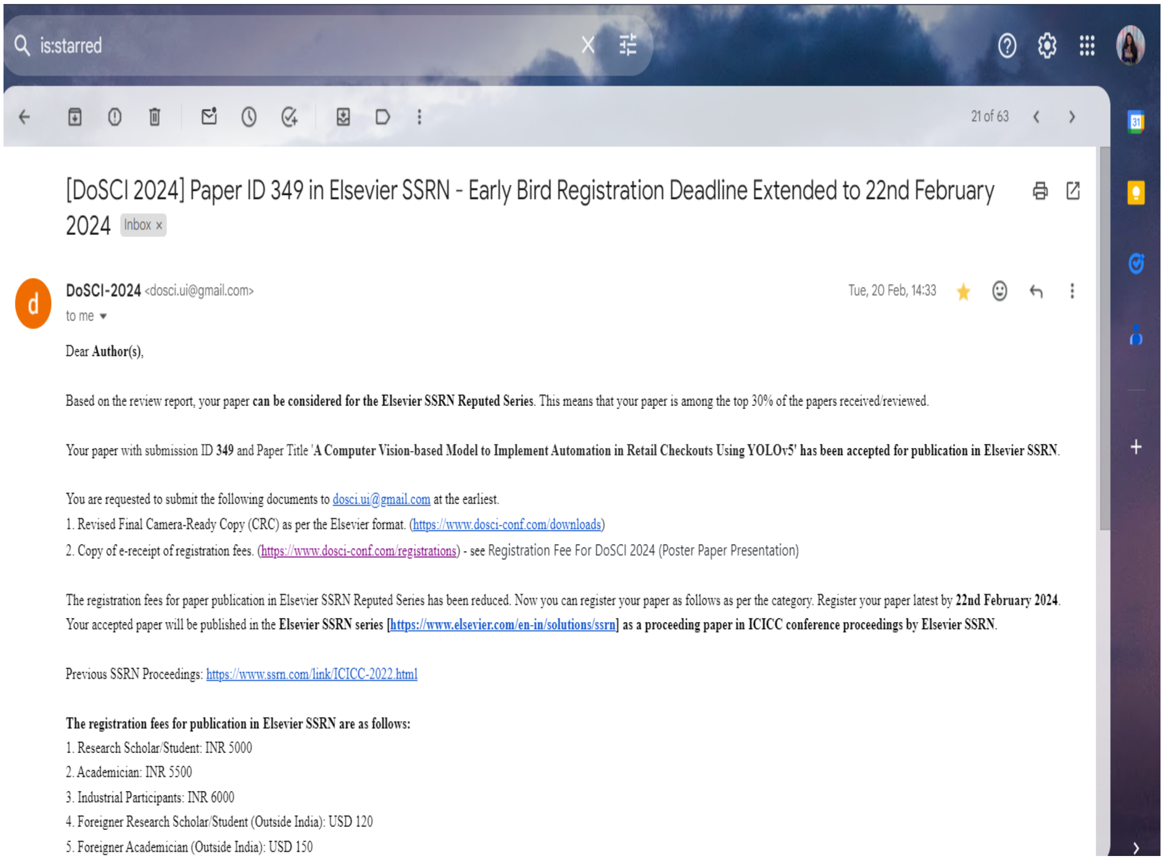
[15] "One-stage object detection." Accessed: Oct. 31, 2023. [Online]. Available: https://machinethink.net/blog/object-detection/

[16] “Software Development Life Cycle (SDLC)” Accessed: Mar. 16, 2024 https://www.javatpoint.com/software-engineering-software-development-life-cycle

[17] “Types of Testing” Accessed: Mar. 16, 2024 https://www.geeksforgeeks.org/types-software-testing/?ref=lbp

Github Link: https://github.com/KIET-Github/CS-2024-C/tree/main/PCS24-30-Nandita

RESEARCH PAPER ACCEPTANCE PROOF



is:starred

21 of 63

[DoSCI 2024] Paper ID 349 in Elsevier SSRN - Early Bird Registration Deadline Extended to 22nd February 2024 Inbox

DoSCI-2024 <dosci.ui@gmail.com>

to me

Tue, 20 Feb, 14:33

Dear **Author(s)**,

Based on the review report, your paper **can be considered for the Elsevier SSRN Reputed Series**. This means that your paper is among the top 30% of the papers received/reviewed.

Your paper with submission ID **349** and Paper Title '**A Computer Vision-based Model to Implement Automation in Retail Checkouts Using YOLOv5' has been accepted for publication in Elsevier SSRN**.

You are requested to submit the following documents to dosci.ui@gmail.com at the earliest.

1. Revised Final Camera-Ready Copy (CRC) as per the Elsevier format. (https://www.dosci-conf.com/downloads)
2. Copy of e-receipt of registration fees. (https://www.dosci-conf.com/registrations) - see Registration Fee For DoSCI 2024 (Poster Paper Presentation)

The registration fees for paper publication in Elsevier SSRN Reputed Series has been reduced. Now you can register your paper as follows as per the category. Register your paper latest by **22nd February 2024**. Your accepted paper will be published in the **Elsevier SSRN series [https://www.elsevier.com/en-in/solutions/ssrn] as a proceeding paper in ICICC conference proceedings by Elsevier SSRN**.

Previous SSRN Proceedings: https://www.ssrn.com/link/ICICC-2022.html

The registration fees for publication in Elsevier SSRN are as follows:

1. Research Scholar/Student: INR 5000
2. Academician: INR 5500
3. Industrial Participants: INR 6000
4. Foreigner Research Scholar/Student (Outside India): USD 120
5. Foreigner Academician (Outside India): USD 150

A Computer Vision-based Model to Implement Automation in Retail Checkouts Using YOLOv5

Anmol Jain[a], Kshiteesh Kumar[b], Nandita Yadav[c], Kumari Bhavya Chaubey[d]

[a]*KIET Group of Institutions, Delhi-NCR Meerut Road, Ghaziabad, 201206, India*
[b]*KIET Group of Institutions, Delhi-NCR Meerut Road, Ghaziabad, 201206, India*
[c]*KIET Group of Institutions, Delhi-NCR Meerut Road, Ghaziabad, 201206, India*
[d]*KIET Group of Institutions, Delhi-NCR Meerut Road, Ghaziabad, 201206, India*

Abstract: With the advent of technology, the retail sector has witnessed significant changes that once used to be believed too unfathomable. Self-service checkout systems have been established in a variety of stores in which customers have to scan all the bought items with a barcode reader and then pay for the bought items. However, these services could result in thefts in these stores and reduce customer experience and impersonal communication between employees and customers. Technologies such as Amazon Go, released by Amazon in 2017, which is based on "Just Walk Out Technology", eased the self-checkout process as the customers do not have to wait in queues to pay for the purchased items. After installing the Amazon Go app, customers scan their iPhones as they enter the store. The track of their purchases is maintained. They can leave the store without physically checking out as their accounts are charged automatically. It makes use of deep learning, computer vision, and sensors. It is built on technology like self-driving cars. However, this technology can hinder customer experience as customers unfamiliar with the app would find it very difficult to shop. Besides the technology used in apps like Amazon Go is expensive. Therefore, to solve these problems and produce an outcome that could favor both the customer and the retailer is described in this paper. Paper focuses on enhancing customer experience by creating a system that can scan the bought items, detect these items with the help of a pre-trained model, and generate a bill containing the total cost that needs to be paid.

Keywords: Computer Vision, Deep Learning, Object Detection, tkinter, YOLOv5

1. Introduction

Nowadays automation techniques have proved to be very beneficial in the retail sector. They play a crucial role in the retail sector by providing better efficiency, reducing costs, enhancing customer experience as well as giving a competitive edge in today's fast-paced market. Many applications make use of Computer Vision techniques to simulate the behavior of human eyesight. The study of Computer Vision allows systems to extract useful data from digital photos, videos, and other forms of media.

One of the steps involved in Computer Vision is Object Detection. Its main objective is to find and localize the objects present in an image or a video. It entails two tasks: Image Classification and Object Localization [1]. While object localization involves locating objects in an image and using a bounding box to denote their locations, image classification identifies the kind or class of an object.

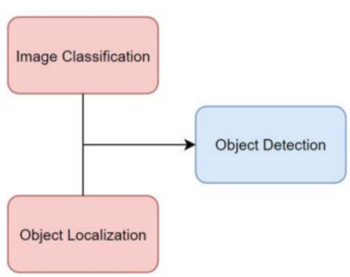


Fig. 1 - Object Detection Tasks

In the case of detecting multiple objects, multiple object detection algorithms are used.

Convolutional Neural Networks (CNN) are popularly used in object detection. Over the past few years, CNN has become famous for all the tasks involved in computer vision as well as object detection [2].

1.1. Types of object detection models:

Object detection methods that are based on CNN, are grouped into two models: One-Stage Models and Two-Stage Models [3]. They are explained as:

A. One-Stage Models:

These models require one pass only through the neural network and predict the bounding boxes in one go. They prioritize inference speed. YOLO series, SSD, and RetinaNet are the representative networks [4].

You Only Look Once (YOLO) is a widely used object detection algorithm, capable of detecting objects in real-time. It has many versions such as YOLOv3, YOLOv5, YOLOv6, etc. SSD (Single Shot Detector) pertains to multi-scale target feature extraction helping in predicting multiple classes. RetinaNet introduces a new function known as the Focal Loss Function that deals with class imbalance during training.

B. Two-Stage Models:

These models consist of two steps: creating multiple proposal bounding boxes in the first phase, and using these bounding boxes to perform item categorization and location information prediction in the second phase [5]. These models prioritize detection accuracy over inference speed. R-CNN, Fast R-CNN, SPPNet, Faster R-CNN, FPN, and Mask-RCNN are the representative models of Two-Stage Models [6].

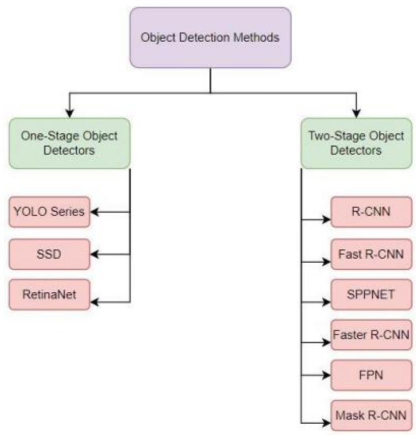


Fig. 2 - Classification of Object Detection Tasks

R-CNN uses selective search approach to propose regions that likely contain objects [7]. Instead of using a selective search algorithm, Fast R-CNN establishes a pooling layer known as Region of Interest (RoI) that extracts fixed-size feature maps from CNN feature maps whereas Faster R-CNN uses an integrated Region Proposal Network (RPN) that shares convolutional features with the detection network [8]. Mask R-CNN extends Faster R-CNN by including a branch for predicting segmentation masks, along with bounding boxes. Spatial Pyramid Pooling Network (SPPNet) is used for arbitrary images without requiring resizing.

Markov Clustering Networks (MCN) and the Viola-Jones technique are used to detect multiple objects and the presence of faces and text within an image. However, these approaches do not account for things like picture size, gradient, orientation, and characters [9].

Our system is based on the YOLOv5 technique of object detection. It is a model in the YOLO series of Computer Vision models and is used to detect multiple objects in an image with fast speed and better accuracy [10]. In this paper, we have discussed the implementation of an automated retail checkout system which generates the bill of the bought products by taking images or videos of the bought products. The main objective of this system is to ease the process of checkout at the retail stores. It can help to reduce the long queues at the checkout counters and can therefore, save customers' time. It aims to enhance the customer-employee relationship and helps to reduce the cashiers' burden to manually scan every item one by one at these stores. It can also be used in maintain the stocks at the retail stores.

The system checks whether the items present in the image/video are present in the dataset or not. For pre-processing the dataset, Roboflow is used. Roboflow has an inbuilt feature that can be used to give annotations. Using it, we put labels on each of the photos. Along with this, Roboflow increases the dataset by rotating every image with different angles. After processing the dataset from Roboflow, we import data in YOLOv5 for further detection. Object Detection and Object Recognition are performed through YOLOv5 which is trained on the COCO Dataset. YOLOv5 is used to achieve better accuracy in object detection. It is trained on custom datasets. The system can then generate an automated bill that includes the information like name of the product, its quantity, and its price. The final amount of all the bought items is also written at the end of the report.

2. Related Work

Many researchers have investigated the applications of Computer Vision in managing retail stores. They have successfully carried out studies and explained the methodologies that could help the retailers manage these stores. A brief literature review is explained in this section.

In 2019, D. A. Mora Hernandez, O. Nalbach, and D. Werth introduced a conceptual tracking system that could generate movement tracks over time for individual customers [11]. In 2020, N. Shekokar, A. Kasat, S. Jain, P. Naringrekar, and M. Shah, prepared an innovative model 'Shop and Go' which makes use of deep learning and sensor fusion [12]. Sensors were used to detect whether the weight had been changed if the item was picked or not. They also stated some real-time applications along with their pros and cons. Figure 3 points to the real-time applications of Self-Checkout Systems.

Fig. 3 - Applications of Self-Checkout Systems [12]

Besides there was also research conducted as to how computer vision can be used to manage product stock at offline retail stores. In 2020, M. A. Majdi, B. Sena Bayu Dewantara, and M. M. Bachtiar proposed a system to find out which goods were nearly empty and misplaced [13]. A camera was used to capture all the displayed products and the products were recognized using YOLOv3, which resulted in achieving the accuracy of 97.61% and 76.67% for misplaced detection. In 2020, C. G. Melek, Elena Battini Sonmez, and Songul Albayrak compared various object detection algorithms and concluded that YOLOv2 is a far better object detection algorithm in terms of both performance and speed for object detection in shelf images [14].

S. K. Yedla, V. M. Manikandan, and V. Panchami in 2020, proposed a novel approach for real-time scene change detection with object detection for automated stock verification [15]. They used a scene change detection technique based on a Structural Similarity Index (SSIM) with the goal of optimizing the processing of video frames.

In the work proposed by M. Sugadev, K. Sucharitha, I. R. Sheeba, and B. Velan, the design and implementation of a hardware-based automated billing system aimed for fruit shops and achieving the Granny Smith Accuracy of 98.896%, was described [16]. It proposed more efficient and accurate billing system than traditional billing systems. They used neural network to classify the fruit and load cell to find the weight of the fruit. In 2020, H. Y. Putra proposed a fraud detection system to detect fraud at self-checkout stores using Data Mining. He predicted fraud using classification techniques and visualized the results to obtain new insight. J48 model had the best performance with F-measure 0.921 [17].

Along with these, there have been research on various object detection algorithms and how do they differ from each other in terms of various factors. The application requirements as well as several other factors, such as speed, accuracy, handling small objects, training complexity, and real-time object detection capacity, influence the choice of object detection method. YOLO excels in real-time object detection as compared to other algorithms. SSD provides a good compromise between speed and accuracy and can handle small objects efficiently.

In Table 1, one-stage object detection methods - SSD and YOLO, and two-stage object detection methods - R-CNN, Fast R-CNN, and Faster R-CNN, are compared based on various parameters like speed, real-time object detection, accuracy in detecting small objects, and training complexity [18].

Table 1 - Comparison of some object detection methods [18]

Point of Difference	R-CNN	Fast R-CNN	Faster R-CNN	SSD	YOLO
Speed	Slow	Medium	Medium	Fast	Fast
Real Time Object Detection	Low	Better than R-CNN	Better than R-CNN and Fast R-CNN	Best	Best
Accuracy in Detecting Small Objects	Low	Better than R-CNN	Better than YOLO	High	Low
Training Complexity	High	Easier than R-CNN	Easier than R-CNN and Fast R-CNN	Easier than two-stage detection methods	Low

3. Proposed Methodology

The proposed system takes video or image as input and generates a bill containing the price and quantity of the bought items. The whole methodology is explained in this section.

The dataset contains 50-50 images of each of the products in various categories. These products include Apples, Oranges, Bananas, Coconut, and Eggs.

We have used Robo Flow, which is a development framework for Computer Vision, used for enhancing data collection to pre-process the data and train the model. It is used to meet the following two objectives:

1) Pre-processing

There is an inbuilt feature in Roboflow for giving annotations. We had put labels on each of the photos. Apart from this, Roboflow also rotated the image from different angles. It was done for every image and hence 2000 approx. images were achieved in comparison to the earlier 720 images.

2) Data Production

In this process, Roboflow divided our dataset into three folders, that is, Training, Testing, and Validation. In each of these folders, there were 2 subfolders - images and labels. Data.yml files extracted all the labels that we gave and then allotted respective labels on respective items. After processing the dataset from Roboflow we received 1 API key and this API key was used to import data in Yolo for further detection.

YOLO is a well-recognized object detection method used widely because of its accuracy and quickness. It is a one-stage approach to object detection. We have used YOLOv5 which is a model trained on the COCO dataset which uses a transfer learning technique [18]. We have tuned this model on our dataset to get better accuracy for object detection. YOLOv5 has five models: yolov5n, yolov5s, yolov5m, yolov5l and yolov5x [19]. In our model, we have used yolov5s which has 7.2 M parameters.

YOLOv5 uses Efficient Det, which is a more complex architecture unlike the previous versions of YOLO. This helps to getter better accuracy and improved generalization of object categories. Apart from this, YOLOv5 is trained on 600 object categories which is a more diverse dataset than other versions. It also utilizes Spatial Pyramid Pooling (SPP) which helps to improve the performance of the model in detecting the smaller objects.

We have used tkinter, which is a standard GUI library for Python, to provide the user interface. The users can upload the images or videos using this interface and the results get displayed.

In Figure 4, we have explained the whole process of checkout at the retail stores with the help of the workflow diagram.

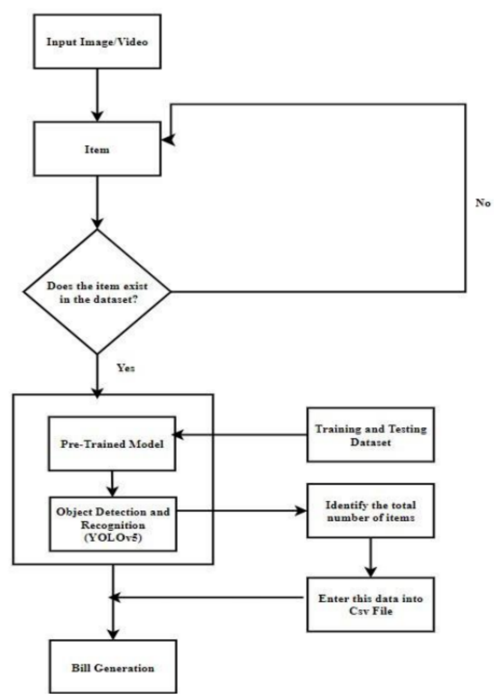


Fig. 4 - Work Flow Diagram of Bill Generation

4. Results and Discussions

When an input image/video containing the items is passed to the system, the system detects whether the data exists in the dataset or not. If yes, then the pre-trained model detects and recognizes the object. Then the detect method, which is made by Tkinter, has a list that contains the per unit cost of each item. The bill is then created in the data frame and the price of each item is calculated. These price values are then added to the data frame and output is displayed.

Figure 5 shows that the model works well in prediction on test samples. The egg is predicted with 70% confidence.

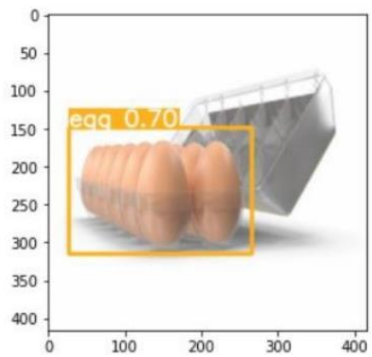


Fig. 5 - Confidence Value for Egg Prediction

When the user clicks on open a file, the interface allows the user to select the image (either in .jpeg or .png format) or video and upload it.

In Figure 6, the input image contains 2 bananas, 2 apples, and 2 oranges. The price of each quantity as mentioned in the list is ['banana': 10, 'apple':50, 'orange':40]. Hence the bill generated is calculated by multiplying the quantity with price and then adding it to the final value.

Fig. 6 - Output (Bill Generation)

5. Conclusion

This system provides an innovative solution that can help the masses to smoothen their checkout experience. It can help the retailers and the customers to effectively checkout using a simple approach. It can generate the bill of the items that contain the information regarding the bought products like name of the time, quantity, weight, price as well as the total bill. The system uses YOLOv5, a very precise and efficient object detection model, to efficiently detect the objects in the video or the image. Since this model uses fewer resources and is cost-effective in comparison to the technologies existing in the market, it can provide an easier approach to checkout services. Retailers as well as customers can benefit from it. It provides a solution to the long queues the customers must stand in to pay the bill.

This system shows great promise but there are certain limitations and areas for further improvement. It is crucial for the retailers to maintain up-to-date datasets for object detection. Apart from this, there might be some errors in accurately identifying and classifying the objects in situations like poor lighting in photos or low-resolution pictures. The future work can be done regarding the improved and novel models for object detection tasks and enhancing the training strategies like transfer learning or domain learning.

Acknowledgement

The authors sincerely thank the editorial board and reviewers for their valuable time and insightful comments.

References

D. Triphena Delight and V. Karunakaran. 2021. "A Comprehensive Analysis of Methodologies used for Object Detection and Localization," in 2021 7th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, Institute of Electrical and Electronics Engineers Inc. pp. 448–453

Y. Hu, X. Li, N. Zhou, L. Yang, L. Peng, and S. Xiao. (2019). "A Sample Update-Based Convolutional Neural Network Framework for Object Detection in Large-Area Remote Sensing Images," IEEE Geosci. Remote Sens. Lett., vol. 16, no. 6, pp. 947–951

"Object Detection: Models, Architectures & Tutorial [2023]." Accessed: Oct. 31, 2023. [Online]. Available: https://www.v7labs.com/blog/object-detection-guide

"One-stage object detection." Accessed: Oct. 31, 2023. [Online]. Available: https://machinethink.net/blog/object-detection/

Anhui da xue et al.,2020, Proceedings of the 32nd Chinese Control and Decision Conference (CCDC 2020), Hefei, China.

"A guide to Two-stage Object Detection: R-CNN, FPN, Mask R-CNN | by Sieun Park | CodeX | Medium." Accessed: Oct. 31, 2023. [Online]. Available: https://medium.com/codex/a-guide-to-two-stage-object-detection-r-cnn-fpn-mask-r-cnn-and-more-54c2e168438c

N. Yadav and U. Binay. (2017). "Comparative Study of Object Detection Algorithms," Int. Res. J. Eng. Technol. [Online]. Available: www.irjet.net

A. John and D. Meva. (2020). "A Comparative Study of Various Object Detection Algorithms and Performance Analysis," Int. J. Comput. Sci. Eng. Open Access Res. Pap., vol. 8, no. 10

Shandong da xue. (2019). Asia-Pacific Signal and Information Processing Association, and Institute of Electrical and Electronics Engineers, 2019 2nd IEEE International Conference on Information Communication and Signal Processing (ICICSP 2019). Weihai, China.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. (2015). "You Only Look Once: Unified, Real-Time Object Detection.", [Online]. Available: http://arxiv.org/abs/1506.02640

D. A. Mora Hernandez, O. Nalbach, and D. Werth. (2019). "How computer vision provides physical retail with a better view on customers," in

Proceedings - 21st IEEE Conference on Business Informatics, CBI 2019, Institute of Electrical and Electronics Engineers Inc., pp. 462–471.

N. Shekokar, A. Kasat, S. Jain, P. Naringrekar, and M. Shah. (2020). “Shop and go: An innovative approach towards shopping using deep learning and computer vision,” in Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Institute of Electrical and Electronics Engineers Inc., pp. 1201–1206.

M. A. Majdi, B. Sena Bayu Dewantara, and M. M. Bachtiar. (2020). “Product stock management using computer vision,” in IES 2020 - International Electronics Symposium: The Role of Autonomous and Intelligent Systems for Human Life and Comfort, Institute of Electrical and Electronics Engineers Inc.

B. Dumnić et al.(2019). IEEE EUROCON 2019 : 18th International Conference on Smart Technologies, Novi Sad, Serbia.

S. K. Yedla, V. M. Manikandan, and V. Panchami. (2020). “Real-Time Scene Change Detection with Object Detection for Automated Stock Verification,” in ICDCS 2020 - 2020 5th International Conference on Devices, Circuits and Systems, Institute of Electrical and Electronics Engineers Inc.

M. Sugadev, K. Sucharitha, I. R. Sheeba, and B. Velan. (2020). “Computer vision based automated billing system for fruit stores,” in Proceedings of the 3rd International Conference on Smart Systems and Inventive Technology, ICSSIT 2020, Institute of Electrical and Electronics Engineers Inc.

H. Y. Putra. (2020). “Fraud detection at self checkout retail using data mining,” in 2020 International Conference on Information Technology Systems and Innovation, ICITSI 2020 - Proceedings, Institute of Electrical and Electronics Engineers Inc.

“What is YOLOv5? A Guide for Beginners.” Accessed: Oct. 31, 2023. [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/

“YOLOv5 - Fine Tuning & Custom Object Detection Training.” Accessed: Oct. 31, 2023. [Online]. Available: https://learnopencv.com/custom-object-detection-training-using-yolov5/

PROOF OF PATENT PUBLICATION

Office of the Controller General of Patents, Designs & Trade Marks
Department of Industrial Policy & Promotion,
Ministry of Commerce & Industry,
Government of India

INTELLECTUAL PROPERTY INDIA

Application Details	
APPLICATION NUMBER	202311057732
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	29/08/2023
APPLICANT NAME	1 . VIKAS KAMRA 2 . Nandita Yadav 3 . Kshiteesh Kumar 4 . Kumari Bhavya Chaubey 5 . Monika Kamra
TITLE OF INVENTION	AUTOMATED RETAIL CHECKOUT SYSTEM USING YOLOV5 IN COMPUTER VISION
FIELD OF INVENTION	ELECTRONICS
E-MAIL (As Per Record)	kamra1984@gmail.com
ADDITIONAL-EMAIL (As Per Record)	kamra1984@gmail.com
E-MAIL (UPDATED Online)	
PRIORITY DATE	
REQUEST FOR EXAMINATION DATE	--
PUBLICATION DATE (U/S 11A)	29/09/2023

Application Status	
APPLICATION STATUS	Awaiting Request for Examination

View Documents

Filed → Published → RQ Filed → Under Examination → Disposed

In case of any discrepancy in status, kindly contact ipo-helpdesk@nic.in