Logo, company name

Description automatically generated

**PUSL3190 Computing Individual Project**

**Project Interim Report**

**Automated self-service system**

Supervisor: Dr. Rasika Ranaweera

Name: Nimesh Hansaka Wanniarachchi

Plymouth Index Number: 10707430

Degree Program: BSc. (Hons) Software Engineering

# Introduction

## Introduction

In the rapidly evolving landscape of the food industry, the combination of automation and image detection technologies carries a huge potential.to streamline process and enhance efficiency. This interim report depicts the intersection of automation and food image detection, aiming to address key challenges and the opportunities in these promises.

As technology develops, particularly artificial intelligence (AI) and computer vision has been heralded a new er[[1]](#footnote-1)a with efficiency and comfort. These technologies provide a huge database to deliver real time operational efficiency and predictive modeling (Kakani et al., 2020)

In 2025,significant changes may occur with AI and robotics in the restaurant business as a survey alone by on that on the implementation of these modification, the restaurant gives a variety of advantages on employees. Convolutional Nural Networks(CNNs) is being a significant computer vision which process image segmentation food authentication and amount estimate.(Hassannejad et al., 2016; Zhang et al., 2015)The research done by Zhang et al.(2020) shows how AI and ML powered for food detection and how system process accurately when identifying variety of food items that captured by cameras. Such kind of advancements not only streamline operations but also minimize human errors. Automated self-service will thoroughly reduce labor costs and enhance the quantity of fields with the safety standards may ultimately be better and more the industry to a new level.(Ajibade et al., 2023)

## Problem Definition

The area that I am facing is the food industry and it faces several challenges including the need for accurate methods of inventory management, and consumer engagement. Traditional manual approaches like food inspection and inventory tracking are often time-consuming, labor intensive and prone to human error. Infact , with the development of the demand for transparency and sustainability, there is a heightened seeking for robust system to ensure product quantity and traceability. (Dinde & Shirgave, 2023)

In the context of self-service restaurants, they are facing challenges in streamlining the ordering, food analysis, and billing processes due to reliance on non-automated systems.

These systems are processed by manual paper-based workflows which lead to inefficiencies, errors, and delays, by consequencing customer dissatisfaction. Manual order taking, handwritten tickets, and manual.

Services which are done in existing restaurants like ordering serving billing can occur in accurate results .

The people who work in restaurants—owners, managers, and employees—bear the brunt of these difficulties because they deal with mistakes and inefficiencies. Self-service restaurants are looking more and more for fully automated systems that use image processing for food recognition, RFID/NFC for tray and customer identification, and QR code systems for quick and safe payment transactions in order to address these problems.(Aguilar et al., 2018)

Through digitization, certain processes can be implemented digitally, improving data transparency, and facilitating more efficient business analysis. Self-service restaurants can lower labor costs, increase customer satisfaction, and improve operational efficiency by automating the billing, food analysis, and payment processes.

## Project Objectives

### Develop automated system for food detection

By training models on enormous datasets of food photos, the initiative aims to automate the process of assessing food by utilizing sophisticated image detection techniques including computer vision and machine learning.

Develop a cutting-edge image processing system that can recognize and categorize food items placed on trays at a self-serve buffet. Sophisticated algorithms will be employed by technology to ensure accurate and efficient food recognition, hence enhancing the dining experience for customers.(Deng et al., 2020)

### Provide food analysis integration with billing systems.

Using an image processing system, a billing system that integrates a food analysis module and calculates cost can be replicated.

The billing system's real-time functioning will guarantee pricing adjustments and open communication between the eatery and its customers.

### RFID Coordination

To easily identify trays, use RFID technology. To establish a unique connection between the tray and the consumer, each tray must have an RFID tag attached to it. RFID integration enhances the customized dining experience, reduces error rates, and increases operational performance.

### QR Code Generations

By reducing wait times at the payment counter, it aims to improve overall customer demands by facilitating rapid and safe transactions.

Include a feature that compiles billing data and creates QR codes for each tray. With these QR codes, customers will have a simple and secure way to complete their payments. Each QR code will contain the total cost of the relevant tray.(Intal et al., 2020)

### Payments Integration

Include a secure payment system to facilitate easy and secure transactions between customers and the restaurant. Include a secure payment system to facilitate easy and secure transactions between customers and the restaurant. Since multiple payment methods are supported with the payment integration, customers will have flexibility and convenience.

### Interface That’s Easy to Use

Make an interface that is simple to use and intuitive for customers and staff at restaurants. The UI of the system is designed to be user-friendly, easily navigable, and to offer a positive experience to all users Use in Self-Serve Restaurants Install the automated buffet system with ease in a range of self-service dining environments. In order to ensure a smooth implementation of the new technology and maximize its benefits, restaurant staff members must get intensive training.

# System Analysis

## Facts Gathering Techniques

Facts Gathering techniques on automatic self-service and food detection process involve several strategies to understand the current landscape and challenges that faced here are some effective techniques.

### Primary

**Observational Studies**

Observed some of the videos of restaurants and supermarkets where there is automated checkout system and food detection technologies in action. It was much better guidance to identify usability issues, technical challenges, and operational workflows.

**Interviews**

Have gone Through interviews with owners of some restaurants. This technique was helpful to gather insights into preferences, experiences regarding the situation and background of processes which happens in a restaurant and what they are willing to have in future to make this process easy.

**Expert Consultation**

Information gathered by discussing with experts in fields of human computer interactions, artificial intelligence and food science was a turning point to direct this project. With their perspectives on technical feasibility, design consideration of automatic self-service system.

### Secondary

**Literature Review**

Existing research papers and articles and reports related to automation in the food industry can be used to gain a huge amount of information about methodologies in detail.

Research by Liu et al.(2016) utilized deep learning techniques in identifying different food items and also the research by Wang ,et al.2005 automatic packing system and how that can be used for future with sensitivity and also the study by wang et al.(2020) explored the implementations of self-service checkout system in supermarket by reducing waiting times for ques.

**Internet**

**YouTube**

## Existing Systems

### Using Computer Vision to Identify Food

The goal of research is to create models and algorithms that can reliably identify and detect different kinds of food. In order to interpret photos and identify various food products, deep learning techniques like convolutional neural networks (CNNs) are used.

The use of deep learning techniques to identify food products kept in a refrigerator was investigated in a paper named "Food Recognition and Detection for Smart Refrigerator Applications" (Gao et al., 2019). The system was designed to help users keep track of food inventories and expiration dates.

A different study, "Food Image Recognition Using Convolutional Neural Network" (Yu et al., 2019), suggested using CNN to identify and categorize food photos that were taken with smartphones. It is possible that the system will be coupled with inventory management or meal ordering systems.(Subhi et al., 2019)

### Food Service Automation

This field of study focuses on automating order taking, preparation, and billing, among other areas of food service. The goals of automation are to increase productivity, decrease human error, and streamline procedures.

A touchscreen kiosk-based system for automating the ordering and billing procedure in fast-food restaurants was presented in "Automated Ordering and Billing System " . Without assistance from a human, customers can place orders and make payments.(Gao et al., 2019)

Researchers created a robotic system that can prepare meals on its own in a study titled "Robotic Kitchen: A Fully Automated Robotic Kitchen Assistant for Assisted Meal Preparation" (Birk et al., 2020). To perform a variety of cooking chores, the device combined robotic manipulation with food detecting skills.

### Combining Food Detection with Invoicing Software

Research looks on how to combine billing systems and food detecting technologies to make checkout quick and easy.

A smart shopping cart with sensors and computer vision technology to recognize objects placed in the cart was proposed in "Smart Shopping Cart with Automated Billing System Using IoT" (Mishra et al., 2020). There is no need for a manual checkout process because the bill is generated automatically by the system.

An automated checkout system that uses computer vision algorithms to detect and recognize products placed on a conveyor belt was introduced in a research titled "Automated Checkout System Using Computer Vision and Machine Learning" (Chen et al., 2019). The bill is then generated by the system using the goods that it has discovered.

### RFID and NFC-Powered Food Monitoring Devices

Research examines how food products can be tracked from manufacturing to consumption using Near Field Communication (NFC) and Radio-Frequency Identification (RFID) technologies. In order to automate checkout procedures and guarantee accurate invoicing based on item identification, these systems can be connected with billing systems.

An RFID-based system to track food products along the supply chain was proposed in "RFID-Based Food Traceability System for Supply Chain Management" (Zhang et al., 2018). By linking RFID tags to product details and prices, the method enables automatic billing.

The article "Mobile Application for Automated Food Ordering and Billing" (Gupta et al., 2019) describes a smartphone application that recognizes food products through image recognition. Through the app, users may directly place orders, with billing determined by the items selected automatically.

### Smart Fridges Using Sensors and Automated Billing

The possibility of automated food identification and billing exists with smart refrigerators that are outfitted with cameras and sensors. The creation of such systems to track food inventories, identify expiring products, and enable automated replenishment billing is being researched.

A smart fridge prototype that uses computer vision and IoT sensors to track food items stored inside was shown in "Smart Fridge with Automated Billing System Using IoT" (Singh et al., 2021). For things that are eaten, the system automatically creates bills and recommends replenishment.

## Use Case Diagram

## Drawbacks of existing systems

### Limited Integration of Technologies

Existing systems often utilize technology in silos, lacking a comprehensive integration that leverages the full potential of AI, ML, and mobile platforms in a cohesive manner. This disjointed approach can limit the effectiveness and accessibility of the technology for all stakeholders in the agriculture sector.

### Accessibility and Usability Issues

Many current agricultural technology solutions are not designed with the end-user in mind, particularly smallholder farmers who may have limited technical literacy. The complexity or lack of user-friendly interfaces can deter farmers from adopting these technologies, thereby limiting their benefits.

### Insufficient Real-Time Data Utilization:

While some systems collect and store agricultural data, they fail to leverage this data effectively for real-time decision-making. The lack of timely insights can prevent farmers from making informed decisions that could enhance their productivity and profitability.

### Inadequate Direct Market Access

Many existing platforms do not adequately address the challenges posed by middlemen in the agricultural supply chain, failing to provide farmers with direct access to markets and fair pricing for their produce. This can result in reduced income for farmers and inefficiencies in the supply chain.

### Data Security and Privacy Concerns

With the increasing digitization of agricultural data, concerns about data security and privacy have emerged. Existing systems may not offer robust protections for the sensitive information of farmers and other stakeholders, potentially deterring participation.

### High Operational Costs and Maintenance

The cost of implementing and maintaining some of the existing agricultural technology solutions such as integrating IOT can be prohibitively high for smallholder farmers and developing economies. This financial barrier can limit the adoption and scalability of such systems.

### Lack of Comprehensive Support Systems

Support systems for troubleshooting, technical assistance, and knowledge sharing are often inadequate in existing solutions, leaving users without the necessary support to fully utilize the technology.

# Chapter 03 – Requirements Specifications

## Functional Requirements

### Automated Food Evaluation

Convolutional neural networks (CNNs) are one sort of image processing technology that the system should use to accurately detect and analyze the various types of food items placed on a tray.

It must be able to handle a wide range of food products with varying sizes, shapes, colors, and configurations so that each is accurately identified.

Real-time results from the food analysis module should enable patrons and restaurant employees to view the flagged items quickly.

### Connectivity to the Billing System

To enable correct cost calculation based on the food items recognized by image processing, the system should smoothly interact with the restaurant's billing system.

Any modifications made by restaurant management should be reflected in the system's dynamic updating of prices and menu items.

Customers should receive itemized bills or receipts from the billing system that include information about every food item identified throughout the self-service procedure.

### Tray and Identification of Customers

The system ought to utilize RFID/NFC technology to provide each tray with a distinct identity and associate it with the corresponding consumer.

Orders and payments made by each customer should be automatically matched to their tray through an error-free identification mechanism.For a more customized eating experience, customers have to be able to link their payment data, loyalty program information, or other preferences to their tray.

### Integration of QR Code Payments

The system ought to produce QR codes with payment information, such as the total amount owed and transaction data.

To avoid fraud or tampering, QR codes have to be properly encoded and specific to each transaction.

Consumers should be able to start the payment process by using their mobile devices to scan the QR code. Secure payment gateway integration should be used to process the payment.

## Non-Functional Requirements

### Achievement

In order to reduce client, wait times, the system should be able to conduct food analysis and invoicing transactions quickly.

It ought to be tuned to manage busy times and large numbers of transactions without lagging or experiencing delays in performance.

Dependability:

During restaurant business hours, the system ought to be dependable and accessible with minimal downtime for upgrades or maintenance.

Fault-tolerant methods should be included to handle mistakes gently and guarantee that clients receive continued service.

### Safety

To safeguard sensitive client data and transaction information, such as payment details and personal information, strong encryption should be used.

To prevent unauthorized users from accessing the system and to protect user privacy, access control procedures should be put in place.

To protect consumer data, the system must abide by applicable data protection laws, such as GDPR or PCI DSS.

### Usability

Both restaurant employees and patrons should find the user interface to be simple to use and intuitive.

From placing orders to making payments, customers should be able to effortlessly complete the self-service process without help. It should be easy for restaurant employees to operate the system with minimal training requirements if they have access to clear instructions and user interfaces.

### Scalability

As the restaurant expands and can handle more patrons and transactions, the system ought to be built to scale easily.

Long-term scalability and flexibility should be ensured by its capacity to accommodate new features or functionalities without requiring substantial modifications to the underlying architecture.

Sustainability:

Modular components and comprehensive documentation should make it simple to maintain and update the system and enable continuous maintenance tasks.

Upgrades and improvements ought to be implemented seamlessly, causing the least amount of disturbance to the system's functioning and leaving no impact on the user's experience.

To guarantee optimal performance and dependability, regular testing and monitoring should be carried out in order to spot any problems early on and take appropriate action.

## Hardware Requirements



1

2

3

4

5

1. Web Camera: Web cameras are a crucial piece of gear that the self-service restaurant system uses to take pictures of the food that is placed on trays. The image processing module uses the visual input from these cameras to precisely evaluate and identify different food items.

Specification: Every self-service station will have HD webcams installed, with a minimum resolution of 720p. To take crisp, detailed pictures of food products, cameras should have a large field of vision and autofocus capability.

2. Mobile Camera: Using their cellphones to scan QR codes and communicate with the self-service restaurant system, consumers can use mobile cameras as extra image capturing devices. The digital interface and the real world can seamlessly integrate thanks to these cameras.

Specification: To optimize user accessibility, compatibility with popular smartphone models (iOS and Android) will be guaranteed. High-resolution picture capture and QR code recognition should be supported by cameras.

3. Plastic Tray: During the self-service dining experience, plastic trays act as the actual platform for holding food items. These trays have RFID/NFC tags installed for tracking and customer identification. This allows for proper billing inside the system and individualized service.

Specification: RFID/NFC tags integrated into lightweight, sturdy plastic trays will be used. Trays should be large enough to hold a variety of foods and cutlery while still being comfortable for customers to grasp and ergonomic.

Tray holders are designed to give plastic trays a firm base, so they stay firmly in place throughout the self-service dining experience. The purpose of strategically placing these holders inside the restaurant layout is to maximize client accessibility and convenience.

Specification: Each self-service station will have sturdy tray holders made of reinforced plastic or stainless steel fitted. To avoid spillage or tipping, holders should have non-slip surfaces and ergonomic designs.

5. RFID/NFC Module: Basically, these modules make it easier for customers and trays to be identified in the self-service restaurant system. These modules allow for wireless communication with RFID/NFC tags implanted in plastic trays, allowing for real-time tracking and customer-specific service.

Specification: Industry-standard protocols (such as ISO/IEC 14443 for NFC) will be supported by dependable and small RFID/NFC devices. Secure communication and smooth integration with the primary system architecture should be supported by the modules.

6. Laptop: Synopsis: The self-service restaurant system software is hosted and operated by laptops, which are the main computing devices. These devices offer the processing capacity and networking needed to efficiently handle image processing, invoicing, and UI elements.

Specification: We'll be using high-performance laptops with solid-state drives (SSDs) for quick storage access, enough RAM (8GB or more), and multi-core processors (such the AMD Ryzen 5 or Intel Core i5). Additionally, a laptop should have enough USB ports for external devices and built-in Wi-Fi.

Make sure the hardware specs support the anticipated functionality and performance expectations and are in line with the technical specifications of your project.

## Software Requirements

A group of logos with different colors

Description automatically generated

**1. Image Processing Module**

In the self-service restaurant system, the image processing module is crucial for food analysis and identification. It uses the most recent deep learning methods—YOLOv8 in particular—for tasks including object identification and recognition.

Implementation: The YOLOv8 model is implemented and trained using TensorFlow, a popular machine learning framework. Furthermore, Roboflow is employed to optimize data preparation and augmentation, hence augmenting the precision and resilience of the model.

**2. Communication Module for RFID/NFC**

This module uses RFID/NFC technology to make tray and customer identification easier. It makes it possible for RFID/NFC modules and the main system to communicate seamlessly, guaranteeing precise tracking and customer-specific service.

Software components in charge of creating and overseeing RFID/NFC connectivity will be developed using Node.js. Effective integration with the hardware components will be accomplished by utilizing pertinent libraries and packages that are compatible with Node.js.

**3. QR Code Generation Tool:**

The QR code generation module dynamically creates QR codes providing payment details for customer transactions. In the self-service restaurant system, it guarantees safe encoding and scanning capabilities, enabling speedy and secure payment transactions.

QR codes is generated programmatically using Node.js-based libraries like qrcode or qr-image. In order to create and display QR codes throughout the payment process with ease, these modules will be easily integrated into the system architecture.

**4. Module for Billing and Payment:**

All billing-related tasks, such as itemized receipts, transaction computations, and payment gateway integration, are handled by the billing and payment module. For consumer transactions, it guarantees safe payment processing and precise billing.

Implementation: To enable safe payment processing, the system will relate to either the Stripe API or the PayPal API. Server-side logic for transaction management, billing computations, and payment gateway communication will be developed using Node.js.

**5. Development of User Interfaces:**

The user interface development process entails using React to create user-friendly and responsive interfaces for both restaurant employees and patrons. It guarantees fluid communication and usage on various platforms and devices.

Node.js is used on the server-side to manage requests and communicate with back-end services, while React is used for front-end development. To facilitate communication between the server-side functionality and the React-based front end, Node.js will be used to construct RESTful APIs.

**6. Integrated Development Environment (IDE):**

Described as a comprehensive toolkit for software development, an IDE offers strong support for JavaScript, Node.js, and React development. Implementation: The main IDEs for software development will be Visual Studio Code or Atom. These IDEs provide a large selection of tools and extensions to improve code quality, productivity, and teamwork throughout the development process.

# Chapter 04 – Feasibility Study

## Operational Feasibility Study

**Evaluation of Current Operations**

As an illustration, carefully examine how self-service restaurants now operate, paying particular attention to the manual ordering procedure, paper-based billing, and payment options. Determine inefficiencies that cause customer discontent because of manual processes, billing problems, lengthy wait times during peak hours, etc.

**Acceptance by Users**

I have used surveys or interviews to get input on the planned food detection and automation invoicing system from restaurant owners, managers, employees, and patrons.It has some quit good acceleration on users that can adjust to new technology and the system satisfies their expectations and needs.

**Availability of Resources**

For instance: Ascertain whether the necessary hardware (such as cameras, RFID/NFC modules), software (such as image processing algorithms, payment gateway integration), and funding is available for the system's implementation. Determine if the restaurant can afford the one-time setup fees as well as the continuing costs of maintenance.

**Technical Proficiency**

Assess the management teams and employees' technical proficiency in running the suggested system. Assess the need for additional technical staff hire or training in order to assist with system setup and maintenance.

**Integration Difficulties**

Determine the possible obstacles to integrating the food detection and automation billing system with the technologies and procedures already in place in restaurants, like point-of-sale systems. To guarantee flawless operations, evaluate compatibility concerns, data migration needs, and the necessity of seamless integration.

**Adherence to Regulations**

Verify that the suggested system conforms with all applicable laws, rules, and guidelines regarding money transactions, data privacy, and food safety. Examine how regulatory constraints, such as the necessity of secure payment processing and data encryption to safeguard client information, may affect the architecture and functionality of the system.

**Analyzing Risk**

Recognize and evaluate potential risks and impediments, including technological risks, market competition, and unanticipated occurrences, that may affect the system's successful deployment and operation. Create backup plans to reduce risks and guarantee company operations.

**Benefit-Cost Analysis**

To ascertain the financial feasibility and possible return on investment of putting the food detection and automation invoicing system into place, perform a cost-benefit analysis. Examine the anticipated benefits, such as enhanced customer satisfaction, cost savings, and higher efficiency, against the estimated costs of development, implementation, and maintenance.

The adoption of the food detection and automation billing system in self-service restaurants can be decided upon by stakeholders after a full assessment of these operational feasibility considerations and resolution of any issues or problems.

## Technical Feasibility Study

Technical viability evaluates a project's suitability in light of available resources and advancements in technology. An example of this would be the implementation of an automated invoicing system and food detection system in self-service restaurants. The following are important elements of technological feasibility, accompanied by examples:

**Technology Evaluation**

Assess whether the selected technologies—such as web development frameworks like React and Node.js, RFID/NFC modules, payment gateways like Stripe and PayPal, and image processing algorithms like Roboflow and YOLOv8—are appropriate for the suggested solution. Make sure that these technologies can function as a cohesive unit and fulfill the project's needs.

**Requirements for Hardware and Software**

List the hardware and software components—such as cameras for picture taking, RFID/NFC modules for identifying customers, servers for database hosting and payment processing, and software development tools—that are required for the system's implementation. Examine these components' compatibility and availability with the current infrastructure.

**Scalability**

As an illustration, think about how scalable the suggested system is to handle future expansion and rising demand. Examine whether a rising user base, more transactions, and a larger volume of data can be handled by the selected technologies and architecture without experiencing appreciable performance degradation. By creating flexible, modular parts that are simple to grow or update, you can plan for scalability.

**Information Administration**

When it comes to information administration, consider the needs for the system's data processing, management, and storage. Ascertain the kind and amount of data—such as menu items, customer orders, transaction records, and billing information—that are produced by the

food detection and billing procedures. To guarantee data security and integrity, determine whether relational or non-relational databases, data encryption, and backup options are required.

**Complexity of Integration**

Evaluate how difficult it will be to integrate the different parts of the system, such as the user interfaces, payment gateways, RFID/NFC communication, QR code creation, and image processing modules. Consider the requirement for bespoke integration solutions, SDKs, or APIs to enable smooth data exchange and communication between various modules.

**Resources for Development**

Assess the availability of qualified engineers, developers, and technical specialists needed to design, build, and implement the system. Determine whether the current team has the requisite experience in software development, systems integration, computer vision, and machine learning. Assess the need for further training or resource hire in order to close any skill shortages.

**Risks Associated with Technology**

As an illustration, list the possible dangers and difficulties that could arise from adopting and using new technologies, such as incompatibilities, software problems, hardware malfunctions, and security flaws. Create risk-reduction plans that include comprehensive testing, prototyping, and continuing support, and maintenance.

Stakeholders can ascertain the technical viability of deploying the food detection and automation billing system in self-service restaurants by carefully evaluating these technical feasibility considerations and resolving any issues or problems.

## Outline Budget

Hardware Costs:

Web Cameras: LKR 10,000

Plastic Trays with RFID/NFC Tags: LKR 5,000

RFID/NFC Modules: LKR 10,000

Total Budget: LKR 25,000

# Chapter 05 - System Architectural Diagram

## Class Diagram

## ER Diagram

## High-Level Architectural Diagram

## Networking Diagram

# Chapter 06 - Development Tools and Techniques

## Development Methodology

This section outlines the methodology employed in the research for the development of an automatic billing system. The study encompasses three essential components that must be accomplished to ensure the completion of the overall project.

**1.Food Detection Model**

**Design and Implementation of Wood-based Stable Holder**

A sturdy and versatile holder was designed to securely hold the tray during data collection. Crafted from wood, it features a C-shaped design for a firm grip. It has been modified to accommodate a webcam, allowing flexibility in camera options. This wood-based holder offers stability, reliability, and adaptability, facilitating high-quality image capture for the research dataset.

A computer and a tray with a blue light

Description automatically generated

**Image Data Set Collection**

The image data set collection plays a pivotal role in training models for object detection and classification. It is crucial to curate a diverse range of photos that accurately represent a wide array of items. This stage holds immense importance as the success of the model heavily relies on the quality and diversity of the data set. To ensure optimal model performance, the data capturing environment should closely resemble the real-life deployment conditions. In this research, the assumption was made that the food items would be placed on a black tray, while the camera would be positioned at a fixed distance and calibrated with appropriate exposure, aided by a stable holder.

For this experiment, ten distinct types of Sri Lankan food items were carefully chosen to compose the data set. These include Cutlet, Donut, Egg roll, Fish Bun, Pastry, Patis, Sandwich, Sausage Bun, Spanchi, and Kibula Bun. These pastries were selected due to their popularity and recognition within the country.

Each pastry variant exhibits unique characteristics in terms of shape, texture, and composition, contributing to the diversity of the data set. However, it is worth noting that certain items may share similar textures or shapes, necessitating precise differentiation during the training process.

To visualize the assortment of pastry items utilized for training the model, Figure 10 provides a comprehensive depiction of each distinct pastry category. The inclusion of these visually distinct images enhances the model's ability to accurately classify and detect the respective pastries during inference. By incorporating a wide variety of pastry types and capturing them from different angles and lighting conditions, the image data set collection ensures robust and comprehensive training, leading to improved performance and reliable results in object detection and classification tasks.

In terms of the implemented dataset, it comprises a total of 2000 photos captured using a camera. The original photo size is 3120px by 4160px. However, to optimize the operational speed on the laptop, each photo was resized to a resolution of 416 X 416 pixels. This reduction in size does not compromise the quality or integrity of the images but rather enhances the efficiency of the model during training and inference phases.

A screenshot of a computer

Description automatically generated

**Labelling the image data set**

Labelling the image data set is a crucial step in computer vision research. In this study, a comprehensive approach was adopted to label the gathered image data set. After the initial organization of the images, each image was individually labeled using the Roboflow tool. This tool facilitated the annotation process by providing a user-friendly interface and efficient annotation capabilities. The resulting annotations were saved in a suitable format for further analysis and model development. The use of computer vision and natural language processing techniques enabled accurate and descriptive annotations that can be processed by computers. The annotation files generated through this process contained valuable information about the objects, features, or regions of interest present in the images. This annotated image data set serves as a vital resource for training and evaluating computer vision algorithms and models. With a total of 2000 labeled files, this data set offers a diverse range of images, contributing to the advancement of computer vision research and applications.

**Figure 11** provides a visual representation of the Labeling annotation interface, showcasing the intuitive and efficient nature of the tool. By leveraging the capabilities of the Roboflow tool and following a systematic annotation process, this research has successfully created a high-quality labeled image data set, paving the way for more accurate and robust computer vision systems.

A screenshot of a computer

Description automatically generated

Figure : Annotating the image data set using Roboflow tool.

Thereafter the annotated images were added to the dataset and classified into 3 sections: train set, valid set and test set.

A screenshot of a computer

Description automatically generated

Then the images were preprocessed using the Auto orient and Resize options available in Roboflow.

A screenshot of a computer

Description automatically generated

Then the augmentation method was used to multiply the number of images. Here the 6 options namely Flip, 90 Rotate, Rotation, Grayscale, Brightness and Blur available in Roboflow were used for this.

A screenshot of a computer

Description automatically generated

Then the images were generated and exported to YOLOv8 with PyTorch library. The generated code link was then copied.

A screenshot of a computer

Description automatically generated

**Choosing the development environment to train the model .**

Choosing the right development environment is crucial for training an effective object detection model. Due to my laptop's limited performance, I opted for an alternative solution. Google Colab emerged as the preferred choice due to its cloud-based platform and powerful computational resources, including high-performance GPUs. This decision aimed to address the demanding computational requirements of training a complex object detection model. Additionally, the researcher leveraged the PyTorch library within the Google Colab environment. PyTorch is known for its flexibility and ease of use, making it ideal for developing and training object detection models. Google Colab provided the necessary infrastructure to handle the computational workload, ensuring faster training and preventing performance bottlenecks. The combination of Google Colab and PyTorch allowed the researcher to efficiently develop and train the model, leading to increased productivity and the ability to conduct extensive experimentation. Ultimately, this approach enabled the researcher to overcome the limitations of my laptop's performance and achieve a highly accurate object detection model for my research project.

A screenshot of a computer

Description automatically generated

**Choosing the algorithm and Training Food detection model**

Choosing the right algorithm and effectively training the pastry detection model are crucial steps in the research process. After completing annotations, the researcher evaluated different algorithms and determined that yolov8 was the most suitable choice due to its exceptional performance and accuracy in object detection.(Basri et al., 2018) By training the model with a dataset of 2000 images, significant results were achieved. The best model satisfied the following requirements.

1.All items in the input image are precisely identified with overlapping and with good accuracy.

2.Identified their locations correctly.

3.Objects that are not in the input image are ignored, increasing the operation speed.

To train the AI model, I utilized yolov8 in conjunction with a dataset comprising 2000 photos representing 10 different Food items. The training process was conducted using the Python programming language, leveraging the capabilities of the PyTorch machine learning platform. This choice of development environment allowed for seamless integration with the yolov8 algorithm and facilitated efficient model training.

Through rigorous experimentation and fine-tuning, the researcher optimized the food detection model's performance, ensuring robustness and high accuracy. The combination of yolov8, the extensive dataset, and the utilization of the PyTorch machine learning framework within a Google Colab environment contributed to the success of the research endeavor. By adopting this approach, I achieved a state-of-the-art food detection model capable of all items in the input image being precisely identified with overlapping and good accuracy, localizing them precisely, and disregarding irrelevant objects for improved operational efficiency.

A screenshot of a computer

Description automatically generated

**Validation the AI model**

Validating the AI model is crucial for assessing its performance and reliability. After training, the model is evaluated using a separate testing dataset. Two types of validation are employed: in sample validation, testing with the same dataset used for training, and out-of-sample validation, using new data.

In-sample validation: Data from the same dataset used to generate the model are tested.

Out-of-sample validation: Validating data from a new dataset that wasn't utilized to create the model (outside of a sample)

By analyzing the model's performance and identifying areas for improvement, informed decisions can be made for its deployment. Effective validation ensures the model's effectiveness and enhances its reliability for real-world applications.

A screenshot of a computer

Description automatically generated

**2.Development of Graphical User Interface**

In my research, I focused on developing a user-friendly graphical user interface (GUI) using the tkinter Python library for the billing system. The tkinter library, being the standard GUI supported by Python, enabled quick and easy GUI creation. The GUI was designed to be intuitive and convenient for users, regardless of their technical background. By leveraging the features of tkinter, I created interactive elements such as buttons and input fields. The GUI incorporated visually appealing design elements and ensured compatibility across different platforms and screen sizes. The development process prioritized user needs, resulting in a well-designed GUI that enhances the usability and accessibility of the billing system. Overall, the tkinter-based GUI provides a seamless and efficient solution for the billing system, contributing to a positive user experience. (Bezerra Beniz & Espíndola, 2016)

A computer screen shot of a program code

Description automatically generated

A screen shot of a computer

Description automatically generated

**React frontend.**

A screen shot of a pizza

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## 3.Final Output of the Billing System

A diagram of a model

Description automatically generated

## Programming Languages and Tools

Several programming languages and tools are used in the creation of the Automated Food Detection and Billing System in order to build various components and features. An overview of the tools and programming languages utilized is provided below:

Python   
Python is a flexible programming language that is frequently used in web development, image processing, and machine learning.   
The backend logic of the system, which includes server-side functionality, object detection models, and image processing algorithms, is implemented in Python.   
  
  
JavaScript   
Programming languages like JavaScript are frequently utilized in web development, especially for creating dynamic and interactive user interfaces.   
Frontend development uses JavaScript to create the web application's user interface for customers and staff members of the restaurant.

React

A JavaScript package called React is used to create user interfaces, especially for single-page applications.   
The web application's frontend is developed using React, which offers a responsive and user-friendly interface for system interaction.

Node.js  
A runtime environment called Node.js enables JavaScript code to be executed on the server side.   
Node.js is used to create server-side parts of the system, like backend and frontend APIs for communication, and to integrate external services.

Flask

A lightweight Python web framework ideal for creating web apps and APIs is called Flask.   
The system's backend server is developed using Flask, which offers APIs for handling queries pertaining to food detection, billing, and payment processing.   
PyTorch and TensorFlow:   
  
Two well-liked machine learning frameworks for creating and honing deep learning models are TensorFlow and PyTorch.   
Convolutional neural networks (CNNs) are implemented and trained in the system for object detection and image processing tasks using TensorFlow and PyTorch.

CSS and HTML   
Web pages are built using HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets), which are essential technologies for determining their layout and style.   
The web application's information is organized and the user interface is styled using HTML and CSS, resulting in a visually pleasing and unified design.

GitHub/Git  
Git is a version control tool used for team collaboration and recording changes made to the source code.   
GitHub is an online platform designed to manage project development and host Git repositories. It facilitates code review, bug tracking, and teamwork.

These programming languages and tools serve as the basis for creating and implementing the Automated Food Detection and Billing System, facilitating the effective execution of all of its features.

## Third Party Components and Libraries

OpenCV

A well-known open-source computer vision toolkit, OpenCV offers a vast array of features for processing images and videos.

Food detection in the system is made more accurate and efficient by using OpenCV for tasks like object detection, feature extraction, and image preparation.

Roboflow

A platform called Roboflow is used to organize, label, and prepare image collections for use in machine learning applications.

The object detection model is trained on an image dataset annotated and preprocessed using Roboflow, which also makes data transformation, normalization, and augmentation easier.

TensorFlow Object Detection API

A framework for developing and implementing object identification models is called TensorFlow Object identification API, and it is built on top of TensorFlow.

Convolutional neural networks (CNNs) are trained and implemented using the Object Detection API to identify and distinguish different food items in system-captured photos.

React Bootstrap

For React apps, React Bootstrap is a UI framework that offers responsive component and layout designs already made.

The web application's frontend incorporates React Bootstrap components to produce an aesthetically pleasing and user-friendly interface for both restaurant staff and customers.

Flask-CORS

For managing Cross-Origin Resource Sharing (CORS) in web applications, Flask-CORS is an extension for Flask.

Requests from the React frontend can be handled by the Flask backend thanks to Flask-CORS, which facilitates communication between the frontend and backend parts of the system.

Stripe API

A framework for processing payments that facilitates safe online transactions and billing administration is called the Stripe API.

By integrating the system with the Stripe API, payments from customers can be accepted and transactions can be processed with ease, guaranteeing a safe and easy billing experience.

QR Code Libraries (e.g., qrcode, qr-image)

Libraries utilized on the backend to dynamically generate QR codes. These libraries create QR codes that may be scanned by mobile devices to start payments by encoding transaction and payment data.

PayPal API

Another payment processing option that makes online payments and financial transactions easier is the PayPal API.

Customers now have an additional payment option thanks to integration with the PayPal API, which streamlines and expands the billing process.

MongoDB

structured and unstructured data is stored and managed in a NoSQL database. MongoDB is scalable and flexible, which makes it appropriate for applications whose data needs change over time.

Cloudinary

a cloud-based solution for managing videos and images. For storing, enhancing, and distributing photos in your application, use Cloudinary. It offers functions that are necessary for managing photos in a web application, like image optimization, scaling, and transformation.

GitLab/GitHub

Repositories like GitHub or GitLab are utilized for collaborative source code development and version management of the system.

Team members may work together on the project, monitor changes, and efficiently handle code contributions with GitLab/GitHub repositories.

By facilitating smooth feature integration and guaranteeing a solid and dependable solution, these third-party parts and libraries significantly improve the Automated Food Detection and Billing System's functionality, efficiency, and usability.

## Algorithms

The Automated Food Detection and Billing System employs several algorithms, mostly from the computer vision and machine learning domains, to carry out different functions. The main algorithms at play are as follows:

**YOLOv8: A rendition of "You Only Look Once"**YOLOv8, a cutting-edge object detection technique, is a member of the single-stage object detection model family.   
It is a good fit for applications where efficiency is crucial because of its reputation for speed and accuracy in real-time object detection tasks.   
To enable automatic meal evaluation, the system uses YOLOv8 to recognize and identify various food items put on trays in self-service restaurants.

**CNNs, or convolutional neural networks**

CNNs are a subclass of deep neural networks that are frequently employed for segmentation, classification, and object detection in image processing applications.   
They consist of several convolutional filter layers that extract features from the input images, pooling layers that reduce dimensionality, and fully connected layers that do classification.   
The image processing module of the system uses CNNs to identify particular food items with high accuracy by analyzing food photos taken by web cameras.

**Transfer of Learning**  
Transfer learning is a machine learning technique in which a model that has already been trained is used as a feature extractor or its parameters are adjusted to fit a new job.   
Within the system, transfer learning can be used to improve the performance of pre-trained CNN models—like YOLOv8—in food detection tasks particular to self-service restaurants by fine-tuning them using a specialized dataset of food photos.

**QR Code Generation Algorithms**  
To dynamically construct QR codes with payment information, including the total amount owed and transaction details, QR code generating algorithms are utilized.   
These methods guarantee the confidentiality and readability of the payment data during the payment process by encoding it into a QR code format.   
Customers can use their mobile devices to start the payment process by using the unique QR codes generated by the system's billing module, which is integrated with QR code creation algorithms.   
These algorithms are essential to the Automated Food Detection and Billing System's ability to evaluate food automatically, identify objects with accuracy, and process payments securely.

# Chapter 07 – Discussion

## Results and Discussion

This research project has mainly considered functions which are Food detecting and the bill calculating System. As well as Qr Code will be generated bill for do payment for customer .customer can do payment scanning qr code using their mobile phone . Main functionality is to capture the image of the food items using the camera provided and conduct real-time detection of the image. First acquired image of food tray and the image running through the trained object detection model. In addition, the detection of several food items on the tray produced positive results from the trained model.

During the development of this system, various constraints were implemented to enhance its accuracy. A limit was set on the number of food products that can be accommodated on the tray, and the scaling of items is determined based on the size of the tray and the items placed on it. Precise identification of all foods that can be placed on the tray is ensured, with each selected food item being represented by more than 1/5 of its actual size.

This project aims to utilize yolov8 for fast and real-time object detection. Since there was no existing food items dataset in Sri Lanka with boundary box labels, it was necessary to capture and label the data. To reduce processing time for each image, the collected images were compressed to a size of 416x416 pixels. These algorithms demonstrated good performance even when the food items overlapped. However, when the background of the food image was complex or similar in color to the food items itself, regional suggestions were not satisfactory. To address this, validation images were taken with the food items placed on a black disc, which provided a strong contrast between the object and background.

## Conclusion

In this research, I have developed an object detection model specifically tailored for Sri Lankan food item image identification and billing system software. The initial step involved gathering a comprehensive dataset of 6000 food item images from Restaurants.

In this study, I created an object detection model specifically for the Sri Lankan food item identification and billing system software. I collected a dataset of 6000 food item photos from restaurants and tested multiple CNN models for object detection, ultimately picking one with an operational speed of 48ms. This approach was linked into a billing platform, which made restaurant operations more efficient. In Sri Lanka, the software is the first of its kind that can identify individual food products as well as duplicated items.

The presented automatic machine cashier, which uses object detection, demonstrates a proof-of-concept for computing retail product pricing using computer vision algorithms rather than traditional methods such as barcode scanning, RFID, or manual price calculation by operators.

This technique brings up new opportunities for improving productivity and decreasing human error in checkout systems.

This research makes a substantial contribution to the integration of the CNN model into the billing application, which improves efficiency and accuracy in detecting food products while also providing a solution for Queues in restaurant.

Future research should focus on improving the accuracy of item detection algorithms and expanding the dataset to include a broader range of food kinds.

## Overview of the Interim Report

This depicts automated food detection using image processing techniques and implementation of a QR code which is generated through the billing system. In today’s fast paced environment where efficiency and customer satisfaction are the prominent, automating processes like these can enhance the dining experience any doubt by leveraging technologies such as computer vision and QR codes, this project seeks to streamline food identification and billing system which ultimately improve efficiency and convenience.

## Summary of Report

Since the methodology involves training convolutional neural networks and functions of billing system , it has been significant progress on automatic self service system in the criteria of restaurants and related areas.

## Challenges Faced

Several challenges were arisen during the project, including the light conditions were affected the accuracy of food detection and technical difficulties integration QR code with existing software and it was supposed to do with several kinds of training module and to address these challenges, here significant objectives and refine the system is based on real world usage.

## Future-Plans/Upcoming Work

Looking ahead, the project will focus on further optimizing the image processing algorithms to enhance the speed and accuracy. And additionally, to explore advanced techniques as transfer learning and also aiming to make some features into QR code billing system, such as rewards and offers ultimately, the goal is to create a robust and user-friendly solution for both customer and businessmen.

# References

Aguilar, E., Remeseiro, B., Bolaños, M., & Radeva, P. (2018). Grab, pay, and eat: Semantic food detection for smart restaurants. *IEEE Transactions on Multimedia*, *20*(12), 3266–3275.

Ajibade, S., Dante, D., Sumarriva-Bustinza, J., Calla, M., Cajavilca, V., & Acosta Lopez, E. (2023). *MACHINE LEARNING (ML) IN FOOD PRODUCTION: THE PROSPECTS AND APPLICATIONS*. *7*, 325–341.

Basri, H., Syarif, I., & Sukaridhoto, S. (2018). *Faster R-CNN Implementation Method for Multi-Fruit Detection Using Tensorflow Platform*. <https://doi.org/10.1109/KCIC.2018.8628566>

Bezerra Beniz, D., & Espíndola, A. (2016). *USING TKINTER OF PYTHON TO CREATE GRAPHICAL USER INTERFACE (GUI) FOR SCRIPTS IN LNLS*.

Deng, J., Xuan, X., Wang, W., Li, Z., Yao, H., & Wang, Z. (2020). A review of research on object detection based on deep learning. *Journal of Physics: Conference Series*, *1684*, 012028. <https://doi.org/10.1088/1742-6596/1684/1/012028>

Dinde, S., & Shirgave, S. (2023). Improved Food Traceability for Restaurant customers using Blockchain Technology. *2023 International Conference for Advancement in Technology (ICONAT)*, 1–7. <https://doi.org/10.1109/ICONAT57137.2023.10080182>

Gao, X., Ding, X., Hou, R., & Tao, Y. (2019). Research on Food Recognition of Smart Refrigerator Based on SSD Target Detection Algorithm. In *AICS 2019: Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*. <https://doi.org/10.1145/3349341.3349421>

Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., & Cagnoni, S. (2016). Food image recognition using very deep convolutional networks. *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management*, 41–49.

Intal, G. L., Payas, J. D., Fernandez, L. M., & Domingo, B. M. (2020). Restaurant Information System (RIS) with QR Code to Improve Service Operations of Casual Fine Dining Restaurant. *2020 IEEE 7th International Conference on Industrial Engineering and Applications (ICIEA)*, 1054–1059. <https://doi.org/10.1109/ICIEA49774.2020.9102036>

Kakani, V., Nguyen, V., Kumar, D., Kim, H., & Visweswara Rao, P. (2020). A critical review on computer vision and artificial intelligence in food industry. In *Journal of Agriculture and Food Research* (Vol. 2). <https://doi.org/10.1016/j.jafr.2020.100033>

Subhi, M., Ali, S., & Mohammed, M. (2019). Vision-Based Approaches for Automatic Food Recognition and Dietary Assessment: A Survey. *IEEE Access*, *PP*, 1. <https://doi.org/10.1109/ACCESS.2019.2904519>

Zhang, W., Zhao, D., Gong, W., Li, Z., Lu, Q., & Yang, S. (2015). *Food Image Recognition with Convolutional Neural Networks*. <https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP.2015.139>

1. [↑](#footnote-ref-1)