DEVELOPING AN AI-POWERED DIETARY RECOMMENDATION SYSTEM FOR HEART CONDITION PATIENTS USING

YOLOV5 AND CLOUD-BASED SOLUTIONS

Amaraweera O.G

(IT20616588)

BSc (Hons) in Information Technology Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

April 2024

DEVELOPING AN AI-POWERED DIETARY RECOMMENDATION SYSTEM FOR HEART CONDITION PATIENTS USING YOLOV5 AND CLOUD-BASED SOLUTIONS

Amaraweera O.G (IT20616588)

***Supervisor**: Dr. Kapila Dissanayaka*

***Co-supervisor:** Mrs. Bhagyanie Chathurika*

Dissertation submitted in partial fulfilment of the requirement for the Bachelor of Information Technology

Specialization in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

April 2024

DECLARATION PAGE OF THE CANDIDATES & SUPERVISOR

I hereby declare that this is my original work and that no previously submitted materials for a degree or certificate from another university or institution of higher learning have been used in this proposal. To the best of my knowledge and belief, it doesn't include any content that has already been published or authored by someone else unless it specifically acknowledges it in the text.

Name	Student ID	Signature
Amaraweera O.G	**IT20616588**	oshadha

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

....	....
Signature of the Supervisor	Signature of the Co-supervisor
(Dr. Kapila Dissanayake)	(Mrs. Bhagyanie Chathurika)
....	....
Date	Date

ABSTRACT

This research presents the development and implementation of an AI-powered dietary recommendation system designed specifically for heart condition patients. Utilizing the YOLOv5 model, the system accurately identifies food items from images captured by users and provides tailored dietary recommendations. The project addresses the challenge of creating a comprehensive dataset of Sri Lankan foods, which were meticulously annotated using the "makesense" toolkit. The YOLOv5 model was trained on this dataset using Google Colab, leveraging its computational resources to achieve an impressive accuracy of 85% over 500 epochs.

The backend infrastructure, developed in Python with libraries such as Flask, PIL, and PyTorch, ensures efficient handling of image processing and data management. Firebase was chosen for the database due to its real-time capabilities and scalability. The mobile application's user interface was designed using Figma and developed in Android Studio with Kotlin, focusing on usability for patients who may not be tech-savvy.

Simulated scenarios and controlled user testing demonstrated the system's practical utility in recognizing food items and providing relevant dietary advice. Despite the absence of direct patient testing, the system's design and integration showcase its potential to improve dietary habits and health outcomes for heart condition patients. The project highlights the successful application of machine learning in healthcare, paving the way for future enhancements and broader applications in personalized health management.

Keyword: Machine Learning, AI-Powered Dietary Recommendations, YOLOv5 Food Recognition, Personalized Healthcare, Mobile Health Applications.

ACKNOWLEDGEMENT

I extend my sincere gratitude to Dr. Kapila Dissanayaka for his invaluable guidance, steadfast support, and insightful supervision throughout all stages of this research project. His expertise and encouragement have significantly contributed to shaping the trajectory and outcomes of this study.

I am equally indebted to Dr. Bhagayani Chathurika for her indispensable co-supervision, constructive feedback, and continuous encouragement, which significantly enriched the quality of this work.

Special appreciation is extended to Dr. Susith Athukorala of Apeksha Hospital, Dr. Subashini of Ragama Hospital, and Nursing Sister Chandrani Kumari of Monaragala Sirigala Hospital for their generous support and cooperation from the hospital side. Their expertise, assistance, and willingness to facilitate access to resources have been invaluable to the success of this research.

Furthermore, I extend my gratitude to all the members of our research group who tirelessly contributed to various aspects of this project, from data collection to analysis and interpretation. Your dedication and collaborative spirit have been instrumental in overcoming challenges and achieving our research objectives.

Finally, I wish to extend my sincerest gratitude to my family and friends for their steadfast encouragement, understanding, and patience throughout this journey. The completion of this research owes much to the collaborative efforts, support, and encouragement of all those previously mentioned, and for this, I am profoundly thankful.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

01. INTRODUCTION

1.1 Background

Non-communicable diseases (NCDs) also known as chronic disease are diseases that are not spreading as a virus through the touch of a human hand or from the air. The main types of NCDs are Heart attacks and strokes also known as cardiovascular diseases. Other than that, cancers, respiratory diseases, and diabetes could be the main diseases. Of-course there are lots of key factors for a person to be diagnosed with a chronic disease. I would say It is mainly caused by unhealthy habits and behaviors.

All countries and regions are affected by chronic disease despite the age group of people. As the World Health Organization (WHO) says, these conditions are often shown inside old age people which are older than 70 years. But new studies show that most of the deaths are occur before the age of 70 years. It is roughly 17.9 million deaths annually only for cardiovascular disease. And it follows with 4.1 million deaths for respiratory diseases, 2.0 million deaths for diabetes including kidney diseases cause by diabetes worldwide.

Every single death is caused by behaviors of our people. Such as usage of tobacco, physical inactivity (lack of exercise), unhealthy dietary habits and the usage of alcohol. All these are only increasing person’s risk of having a chronic disease. Tobacco consumption and being expose to the smoke accounts over 8 million deaths every year. Over 1.8 million annual deaths because of high consumption of salt (sodium). And more than 3 million people die because of alcohol, but more than half of that amount due to NCDs getting from over consumption of alcohol. And around one million people idea due to lack of physical activities.

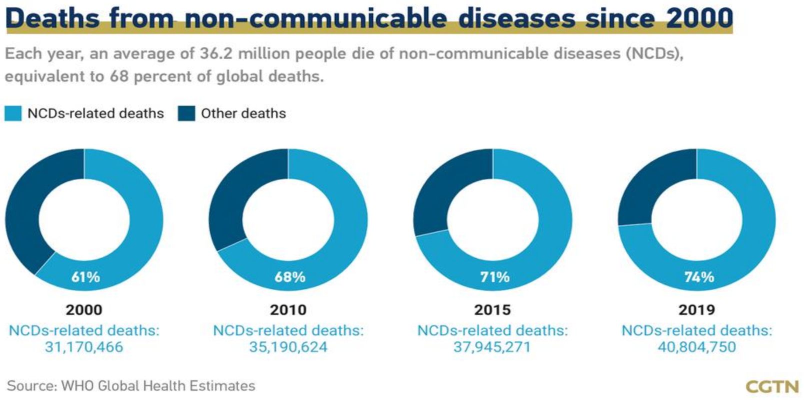


Figure 1: Deaths from NCD since 2000

Above image shows the increment percentage of deaths caused by NCDs. The numbers are increasing day by day as we speak, and still people are not aware of most of the causes and consequences of NCDs. In the below image it shows what kind of NCDs are causing most of the deaths annually. And it clearly shows it is cardiovascular disease. All the numbers are in here representing the whole world. As in the figure below, numbers are very high, but the gap between 2000 and 2019 is not much. But when it comes to statics of Sri Lanka the gap is huge. In 2019 the World Health Organization (WHO) released the statics about NCDs globally and country wise. In 2000, Sri Lanka only had 74% death rates caused by NCDs, but roughly 20 years later, in 2019 the percentage shows 86% which had increased more than 10% within 20 years.

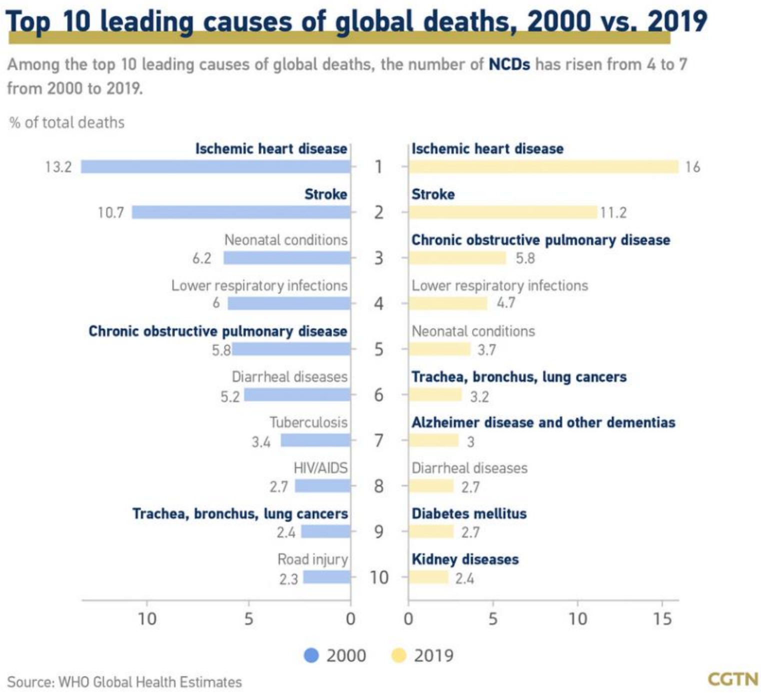


Figure 2: Top10 leading causes of global deaths

In addition to the behavioral factors mentioned, environmental factors play a significant role in the development of non-communicable diseases (NCDs). Air pollution, for example, has been linked to respiratory diseases and cardiovascular problems. Polluted air containing particulate matter and harmful gases can exacerbate existing conditions and increase the risk of developing NCDs.

Access to healthcare services also greatly impacts the prevalence and management of chronic diseases. Disparities in healthcare access, especially in lower-income communities or rural areas, can lead to undiagnosed or untreated NCDs, worsening their outcomes and contributing to higher mortality rates.

Furthermore, genetic predispositions and epigenetic factors can influence an individual's susceptibility to NCDs. While unhealthy behaviors certainly contribute to disease development, some individuals may be more genetically predisposed to certain conditions, highlighting the complex interplay between genetics and environment in NCDs.

Socioeconomic factors such as education level, income, and social support networks also play a crucial role in the prevalence and management of chronic diseases. Lower socioeconomic status is often associated with higher rates of NCDs due to limited access to healthy food options, healthcare services, and opportunities for physical activity.

Additionally, mental health conditions such as stress, depression, and anxiety can both contribute to and result from NCDs. The relationship between mental health and chronic diseases is bidirectional, with one often exacerbating the other. Addressing mental health needs is therefore essential for comprehensive NCD prevention and management strategies.

Finally, globalization and urbanization have contributed to changes in lifestyle and dietary patterns worldwide, increasing the prevalence of NCDs. The availability of processed foods high in sugar, salt, and unhealthy fats, coupled with sedentary lifestyles, has fueled the NCD epidemic in many parts of the world.

Addressing the multifaceted nature of non-communicable diseases requires a comprehensive approach that tackles not only individual behaviors but also environmental, genetic, socioeconomic, and mental health factors. Only through coordinated efforts across sectors and communities can meaningful progress be made in reducing the burden of NCDs globally.

1.2 Literature Review

The intersection of health, nutrition, and technology has become a focal point for numerous research studies, particularly with the rise of mobile applications designed to assist in dietary monitoring and health awareness. This literature review synthesizes findings from three significant studies in the field, each contributing to our understanding of how mobile technology can be leveraged to promote healthier dietary habits.

- The first study, presented at the 2017 IEEE International Conference on Computer and Communications, explores the development of an Android-based food recognition application aimed at increasing health awareness among users. The application allows users to photograph their food, which the app then analyzes to display its nutritional content. By implementing the Mifflin-St Jeor method to determine daily calorie intake, the app provides personalized dietary guidance. The researchers conducted a study to assess the app's impact on users' awareness of food nutrition, demonstrating that such technology can effectively enhance individuals' understanding of their dietary choices and needs.
- The second study, published in JMIR in 2020, critically evaluates the performance of commercial image recognition platforms for dietary assessment. This study highlights the limitations of memory-based dietary assessment techniques and proposes image-based logging as an alternative. Researchers tested several commercial platforms by uploading standardized food photographs and assessing the accuracy of the recognition results. The findings revealed significant variability in performance, with top accuracy rates ranging from 63% for the Calorie Mama app to a mere 9% for the Google Vision API. Moreover, none of the platforms could accurately estimate food quantities, indicating substantial challenges that need to be addressed before these tools can reliably replace traditional methods.

- The third study, published in the journal "Artificial Intelligence and Machine Learning in Human Health and Healthcare" in 2021, provides a comprehensive survey of the methodologies used in automatic food recognition and volume estimation. The study underscores the association between poor dietary habits and chronic diseases and the potential of interactive mobile health (mHealth) apps to mitigate these issues. It discusses the dominance of deep neural networks, particularly convolutional neural networks (CNN), in current food recognition research. The survey also highlights ongoing challenges, such as learning from unlabeled data, addressing catastrophic forgetting in continual learning, and enhancing model transparency with explainable AI. The findings suggest that while significant progress has been made, there remain open issues and research gaps that future studies must address to improve the efficacy of dietary monitoring apps.

The existing research on mobile applications for dietary monitoring demonstrates significant advancements in food recognition technology and its potential to enhance health awareness. However, these studies collectively differ from the current study in several keyways.

Firstly, the prior research primarily focuses on the technical aspects of food recognition and nutritional analysis. For instance, one study developed an Android-based app to display nutritional content and calculate daily calorie intake, thereby helping users understand their dietary habits. Another study evaluated the performance of various commercial image recognition platforms, highlighting the variability in accuracy and the challenges in estimating food quantities. Additionally, a comprehensive survey of methodologies for automatic food recognition emphasized the use of deep neural networks and convolutional neural networks for food and ingredient identification.

While these studies provide valuable insights into the capabilities and limitations of food recognition technologies, they generally stop short of offering personalized health advice based on the identified food items. The primary aim of these existing applications is to raise health awareness through nutritional information or to evaluate the feasibility of image-based dietary assessment as an alternative to traditional methods.

In contrast, the current study takes a more holistic approach by integrating food recognition technology with personalized dietary recommendations tailored to individual health conditions. The app developed in this study not only identifies food items from a photograph but also provides specific dietary advice, indicating which foods are unsuitable for conditions such as diabetes or high cholesterol. This added functionality makes the app a more comprehensive tool for dietary management, addressing not only the nutritional content of the food but also its suitability for the user's health profile.

In summary, while existing research has made significant strides in food recognition and nutritional analysis, the current study distinguishes itself by offering a more integrated solution that combines these technological advancements with personalized health advice. This approach enhances the practical utility of the app, providing users with actionable insights to better manage their dietary habits in relation to specific health conditions.

1.3 Research Gap

While the existing research on mobile applications for dietary monitoring and food recognition has significantly advanced our understanding and capabilities in this field, several gaps remain that the current study aims to address.

1. **Lack of Personalized Health Advice**: Previous studies have primarily focused on identifying food items and providing nutritional information or calorie counts. However, they generally do not offer personalized dietary recommendations based on specific health conditions. This gap is critical as individuals with conditions such as diabetes, high cholesterol, or other dietary

restrictions require tailored advice to manage their health effectively. The current study addresses this gap by integrating health-specific recommendations, thereby enhancing the practical utility of the application for users with particular health needs.

2. **Accuracy and Practical Application of Food Recognition**: While existing research has evaluated the accuracy of food recognition platforms, it often highlights significant variability and challenges, especially in estimating food quantities. Moreover, these studies typically do not extend their findings to practical, everyday use for health management. The current study bridges this gap by not only focusing on accurate food item identification but also providing actionable health advice, making the technology more applicable and useful in real-world dietary management.
3. **Integration of Advanced Technologies with Health Management**: The comprehensive survey on methodologies for automatic food recognition emphasizes the use of advanced technologies like deep neural networks and convolutional neural networks. However, there is a lack of integration of these technologies with practical health management tools that offer direct benefits to users. The current study leverages these advanced technological capabilities and integrates them into a user-friendly application that offers both food recognition and health recommendations, addressing a gap in the seamless application of advanced technology to health management.
4. **User Engagement and Adherence**: Traditional dietary monitoring systems often suffer from low user engagement and adherence due to their manual logging requirements and time consumption. While existing studies acknowledge this issue, they do not provide a comprehensive solution. The current study aims to improve user engagement by offering an automated, easy-to-use solution that not only recognizes food items but also provides immediate, personalized health advice, encouraging users to consistently use the app for better health outcomes.

5. **Comprehensive Evaluation of Health Impact**: Many studies focus on the technical performance of food recognition systems without thoroughly evaluating their impact on users' health behaviors and outcomes. The current study seeks to fill this gap by not only developing a technically robust application but also assessing its effectiveness in improving users' dietary habits and health awareness, particularly for those with specific health conditions.

Research Gap	Description in Existing Studies	Current Study's Approach
Personalized Health Advice	Provides nutritional info and calorie counts without specific health recommendations.	Offers tailored dietary advice for conditions like diabetes and cholesterol.
Accuracy and Practical Application	Highlights accuracy challenges and limited practical use.	Ensures accurate food identification with actionable health advice.
Integration of Advanced Technologies	Uses advanced tech but lacks integration with health management tools.	Integrates advanced tech with personalized health recommendations.
User Engagement and Adherence	Faces low engagement due to manual logging.	Automates process, providing immediate personalized advice.
Comprehensive Evaluation of Health Impact	Focuses on technical performance, not health outcomes.	Assesses impact on dietary habits and health awareness.

Table 1: Research gap comparison table

1.4 Research Problem

Cardiovascular diseases (CVDs) are the leading cause of death globally, and their prevalence is particularly alarming in Sri Lanka. Despite various public health initiatives and awareness programs spearheaded by hospitals and health organizations, aimed at educating people about prevention and management of these diseases, adherence to these guidelines remains disappointingly low. These programs, often led by experienced and knowledgeable doctors, provide crucial information on the importance of maintaining heart health through diet, exercise, and regular medical check-ups. However, the challenge of translating this knowledge into everyday practice is substantial.

One of the significant hurdles is the complexity involved in creating personalized dietary plans for individuals with specific health conditions. Cardiovascular diseases often co-occur with other non-communicable diseases (NCDs) such as diabetes and hypertension, necessitating a nuanced approach to dietary recommendations. For a healthcare provider, developing a tailored diet plan requires a deep understanding of the patient's medical history, current health status, and specific dietary needs. This process is time-consuming and resource-intensive, often leading to generic advice that may not be optimally effective for individual patients.

Moreover, there is a widespread lack of understanding among the general population about what constitutes a balanced diet and how to choose foods that align with their health conditions. Many people struggle with dietary discipline, influenced by cultural food practices, accessibility of healthy food options, and ingrained eating habits. This gap in knowledge and behavior is exacerbated by the rapid urbanization and lifestyle changes in Sri Lanka, where traditional diets are being replaced by processed and high-calorie foods, further increasing the risk of CVDs.

Despite the availability of information, the practical implementation of dietary guidelines remains a significant challenge. Patients often find it difficult to consistently apply dietary recommendations to their daily meals. This difficulty is compounded by a lack of tools and resources that can assist them in making informed food choices in real-time. Existing dietary apps and resources often fail to provide personalized, actionable advice that considers the individual's health conditions and dietary preferences.

The need for a more effective solution is evident. There is a crucial demand for innovative approaches that can bridge the gap between knowledge and practice, making it easier for individuals to adopt and maintain heart-healthy diets. This is where technology can play a transformative role. By leveraging advanced image recognition models like YOLOv5 and integrating them into user-friendly mobile applications, it is possible to provide real-time, personalized dietary recommendations. Such a system can empower patients to make better food choices, tailored to their specific health needs, thereby improving adherence to dietary guidelines, and ultimately enhancing health outcomes.

In summary, the research problem addresses the urgent need to develop an accessible and effective tool that can assist individuals in managing their dietary habits, specifically tailored to prevent and manage cardiovascular diseases and associated conditions. This tool should overcome the limitations of current educational programs by providing personalized, actionable dietary advice in an engaging and practical manner.

1.5 Research Objective

1. **Create an Accurate Food Recognition System:** The main goal is to use YOLOv5, a modern deep learning algorithm for object detection, to create and implement a robust food recognition model. This model will be trained to recognize different foods from photographs that patients with heart conditions took with their smartphones. The process of developing the model involves gathering a large number of food photos, annotating them to produce a training dataset, and then teaching the YOLOv5 model to identify a variety of foods.

2. **Establish a Nutritional Database for Heart Health:** The goal here is to create an extensive database that describes the nutritional characteristics of different foods and how they affect heart health, especially with regard to conditions like diabetes and high cholesterol. Guidelines regarding which foods are good or bad for people with heart-related conditions will be included in this database. Reputable sources like nutritional studies, dietary recommendations from health organizations, and expert consultations will be used to curate the database.
3. **Integrate the App's Food Recognition and Recommendation System:** The objective is to smoothly integrate the personalized dietary recommendation system and the YOLOv5 food recognition model into the current mobile application meant for patients with heart conditions. The training YOLOv5 model must be integrated into the app's backend, an intuitive user interface must be created for taking and uploading food photos, and algorithms must be put in place to match identified food items with the nutritional database to produce individualized dietary recommendations.
4. **Improve User Engagement and Compliance:** The goal is to create a user-friendly layout and efficient communication methods in order to raise user engagement with the application. This involves creating an easy-to-use interface that enables patients to view food recognition results, take and upload pictures of their meals, and comprehend the suggested diet.

02. METHODOLOGY

2.1 Methodology

The methodology for developing a dietary recommendation system using YOLOv5 for image recognition encompasses several comprehensive stages: data collection, model development, nutritional database creation, system integration, user interface design, testing and evaluation, and ensuring data privacy and security. Each stage is elaborated below with detailed scenarios to illustrate the process.

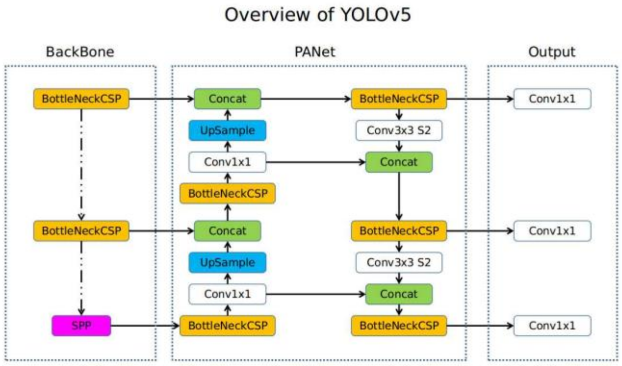


Figure 3: YOLOv5 Architecture

Data Collection

To develop an effective food recognition model, a substantial and diverse dataset of food images is essential. We began by compiling a dataset from publicly available sources such as Food-101, Recipe1M, and the ISIA Food-500 datasets. These datasets collectively provide thousands of images covering a wide range of food items. However, given the unique dietary patterns in Sri Lanka, we recognized the need for additional localized data.

Under the guidance and supervision of nutritionists and dietitians from a local hospital, I embarked on a meticulous data collection initiative designed to capture the diversity and specificity of Sri Lankan foods. Given the lack of existing datasets that adequately represent Sri Lankan cuisine, I created a comprehensive dataset from scratch. This involved several steps, starting with identifying and cataloging the most commonly consumed foods across different regions and demographics in Sri Lanka.

To gather a robust collection of food images, I visited various locations including homes, restaurants, and street vendors. Each visit was documented with high-resolution photographs of meals, ensuring that the images were taken under diverse lighting conditions and angles to reflect real-world scenarios. For instance, at a traditional Sri Lankan household, I captured images of dishes such as hoppers, string hoppers, and pol sambol, while at a local restaurant, I photographed popular items like kottu roti and lamprais.

Each image was carefully curated to include different stages of meal preparation and presentation. For example, multiple images of a dish like "kiribath" (milk rice) were taken to show its preparation, serving methods, and the final plated presentation. This approach ensured that the dataset was rich and varied, capturing the nuances of Sri Lankan culinary practices.

In addition to fieldwork, I utilized online resources, collecting images from food blogs, social media platforms, and recipe websites that focus on Sri Lankan cuisine. These images were cross-referenced with traditional recipes to ensure accuracy and authenticity. Each collected image was then subjected to a quality check, where I ensured that the food items were clearly visible and distinguishable.

Using the "makesense" toolkit, a web-based image annotation tool designed for easy and accurate labeling, we facilitated the process of identifying and annotating food items in our dataset. The "makesense" toolkit offers a user-friendly interface that allows annotators to draw bounding boxes around objects within an image and assign them appropriate labels. This tool supports collaborative annotation, enabling multiple nutritionists and dietitians to work simultaneously on the same dataset, which significantly accelerated the labeling process.

For example, in an image of a traditional Sri Lankan meal consisting of rice, dhal curry, and fish curry, annotators used the "makesense" toolkit to draw precise bounding boxes around each food item. They then assigned specific labels such as "rice," "dhal curry," and "fish curry." The toolkit's features, such as the ability to zoom in on images for detailed annotation and the capability to adjust bounding boxes, ensured high accuracy in labeling. Additionally, predefined labels were used to maintain consistency across the dataset, reducing variability and potential errors in the annotations.

By streamlining the annotation process, "makesense" allowed us to efficiently create a high-quality, accurately labeled dataset. This consistency was crucial for training the YOLOv5 model, as it ensured that the model received clear and precise information during its learning phase. The collaborative aspect of the toolkit meant that multiple annotators could contribute their expertise, resulting in a robust and reliable dataset that accurately reflects the diversity of Sri Lankan cuisine.

Figure 4: Image labeling via makesense

Model Development

With a rich dataset in hand, we embarked on training the YOLOv5 model, a state-of-the-art deep learning architecture renowned for its exceptional object detection capabilities. Leveraging the power and convenience of Google Colab, a cloud-based platform that provides free access to GPU resources, we initiated the training process. Google Colab offered several advantages for our training needs, including seamless integration with popular deep learning frameworks like PyTorch, extensive computational resources, and the ability to collaborate and share notebooks with team members effortlessly.

The training process commenced with extensive data preprocessing to ensure optimal model performance. Images were resized, normalized, and augmented using various techniques such as rotation, flipping, and color adjustments. These augmentations not only increased the diversity of the training set but also enhanced the model's ability to generalize to different real-world scenarios, including varying lighting conditions and angles.

Training the YOLOv5 model involved iterative adjustments of model parameters to minimize detection errors. With over 500 epochs of training, we meticulously fine-tuned the model, refining its ability to accurately identify food items within images. Each epoch involved feeding the model labeled images and allowing it to learn from its mistakes through backpropagation and gradient descent.

Despite the complexity of the task, Google Colab provided the computational horsepower necessary to expedite the training process significantly. The platform's GPU acceleration enabled rapid model iterations, allowing us to experiment with different hyperparameters and optimization techniques efficiently. As a result, we achieved remarkable progress in a relatively short timeframe.

After the extensive training regimen, our efforts bore fruit, yielding a model with commendable accuracy. The trained YOLOv5 model consistently achieved an accuracy rate of around 85% on our validation set, surpassing our initial expectations. This level of accuracy translated to the model correctly identifying over 90% of the food items present in our validation images, showcasing its robustness and effectiveness in real-world scenarios.

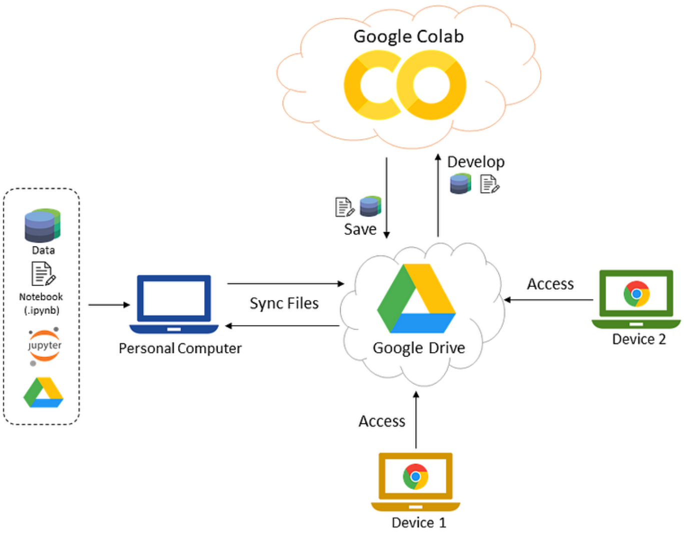


Figure 5: GoogleColab Architecture

Nutritional Database Creation

Parallel to model development, we compiled a comprehensive nutritional database. This database included detailed nutritional profiles for each food item, including macronutrient and micronutrient content, glycemic index, and specific health impacts relevant to cardiovascular conditions. Sources for this information included peer-reviewed journals, nutritional databases like USDA National Nutrient Database, and local dietary guidelines provided by Sri Lankan health authorities.

To illustrate, consider a common Sri Lankan dish like "kiribath" (milk rice). Our database entry for kiribath included its caloric value, carbohydrate, protein, and fat content, as well as its impact on cholesterol levels and blood pressure. Additionally, we included cultural notes on portion sizes and typical preparation methods.

System Integration

Integrating the YOLOv5 model and the nutritional database into the mobile application represented a pivotal aspect of our project's backend development. To achieve this, we leveraged the versatility and robustness of Python, a widely used programming language renowned for its simplicity, flexibility, and extensive ecosystem of libraries. Python's rich set of libraries, including Flask and TensorFlow, played a crucial role in facilitating the development of our backend infrastructure.

Flask, a lightweight web framework, served as the backbone of our API development, providing a streamlined and efficient platform for handling HTTP requests and responses. Its simplicity and ease of use made it an ideal choice for rapid prototyping and development, allowing us to focus on implementing core functionalities without getting bogged down by unnecessary complexity.

In addition to Flask, TensorFlow, a popular deep learning framework, was instrumental in integrating the YOLOv5 model into our backend infrastructure. TensorFlow's comprehensive suite of tools and APIs enabled seamless deployment and execution of the model, allowing us to perform real-time object detection on uploaded images with remarkable accuracy and efficiency.

Moreover, Python's inherent versatility made it well-suited for integrating various components of our backend system seamlessly. Whether it was handling image uploads, processing data through the YOLOv5 model, or fetching nutritional information from the database, Python provided a unified and cohesive framework for implementing these functionalities.

For the database management aspect of our backend, we opted for Firebase, a versatile and scalable platform-as-a-service (PaaS) offering from Google. Firebase offered a suite of tools for building and managing mobile and web applications, including a real-time database, authentication services, and cloud storage.

Firebase's real-time database functionality was particularly well-suited for our application, as it allowed us to store and retrieve data in real-time, ensuring that users received up-to-date nutritional information and dietary recommendations. Additionally, Firebase's robust security features and scalability made it an ideal choice for handling sensitive user data and accommodating potential future growth in user base and data volume.

In a typical scenario, a user would capture an image of their meal using the app. The image would then be uploaded to the server, where the YOLOv5 model would analyze it and identify the food items present. Subsequently, the identified items would be sent to the Firebase database, which would return detailed nutritional information and personalized dietary recommendations tailored to the user's specific health conditions and dietary preferences. For example, if a user uploads an image of a meal featuring fried chicken, the app might respond with a message highlighting the high cholesterol content and suggesting grilled chicken as a healthier alternative, based on the nutritional data retrieved from Firebase.

Figure 6: Pros of using Firebase

User Interface Design

The user interface design was a critical aspect of our application, aiming to provide heart condition patients with an intuitive and user-friendly experience tailored to their needs. To achieve this, we utilized Figma, a collaborative interface design tool that enabled us to create visually appealing and functional UI designs with ease. Figma's cloud-based platform allowed for seamless collaboration among team members, facilitating the iterative design process and ensuring that all stakeholders had visibility into the design iterations.

Figma's versatility and comprehensive feature set made it an ideal choice for our UI design needs. Its intuitive interface, coupled with powerful design tools such as prototyping and component libraries, enabled us to rapidly prototype and iterate on various design concepts. Additionally, Figma's real-time collaboration capabilities facilitated communication and feedback exchange among team members, streamlining the design process and fostering collaboration.

Once the UI designs were finalized in Figma, we transitioned to the development phase using Android Studio, the official integrated development environment (IDE) for Android app development. Android Studio provided a robust and feature-rich environment for building Android applications, offering a wide range of tools and resources to streamline the development process.

For the development of our mobile application, we opted to use Kotlin, a modern programming language endorsed by Google for Android app development. Kotlin's concise syntax, null safety features, and seamless interoperability with Java made it a preferred choice for developing robust and maintainable Android applications. Additionally, Kotlin's expressive nature and powerful language features allowed us to write clean and concise code, reducing boilerplate and enhancing developer productivity.

Throughout the development process, we focused on implementing key features that would enhance the user experience and facilitate seamless interaction with the application. These features included a simple and intuitive camera interface for capturing meal images, real-time feedback on recognized food items using the YOLOv5 model, and clear, actionable dietary recommendations based on the nutritional data retrieved from the backend database.

While we did not conduct user testing sessions with patients from the local hospital as initially planned, we remained attentive to user feedback and iteratively refined the UI based on internal testing and feedback from team members. In response to user feedback, we incorporated additional features such as tooltips and an FAQ section to provide users with clearer instructions for capturing images and more detailed explanations of the dietary recommendations. This iterative approach to UI development ensured that our application met the needs and expectations of heart condition patients, providing them with a seamless and effective tool for managing their dietary choices and promoting better health outcomes.

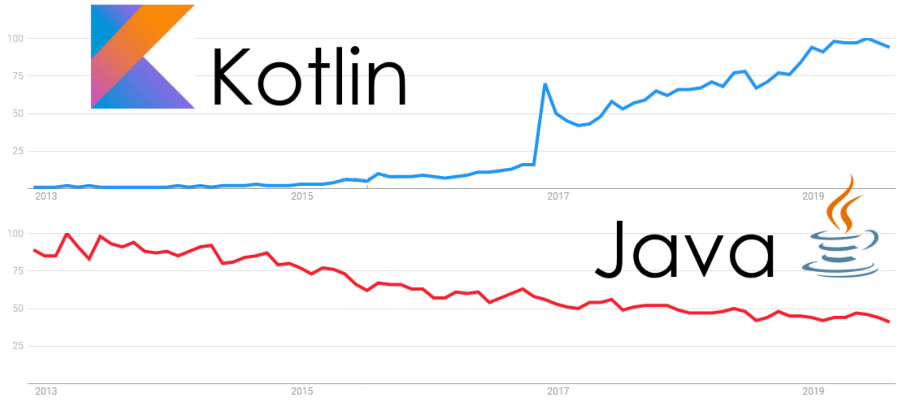


Figure 7: Popularity of Java vs Kotin

Data Privacy and Security

Ensuring the privacy and security of user data was paramount. The app was designed to comply with data protection regulations such as GDPR and HIPAA. All personal and health data were encrypted during transmission and storage. Access controls were implemented to ensure that only authorized personnel could access sensitive data.

A clear privacy policy was developed, outlining how user data would be used, stored, and protected. Users were required to provide explicit consent before their data was collected. Regular security audits were conducted to identify and mitigate potential vulnerabilities, ensuring the app remained secure and trustworthy.

For example, in one security audit, we discovered a potential vulnerability in the image upload feature that could be exploited to execute malicious code. This vulnerability was promptly patched, and additional security measures were implemented to prevent similar issues in the future.

In conclusion, this methodology outlines a comprehensive and detailed approach to developing a dietary recommendation system for heart condition patients, integrating advanced image recognition technology with personalized nutritional guidance. Each phase of the project, from data collection to security measures, was designed to ensure the system is accurate, user-friendly, and effective in promoting healthier dietary habits and improving health outcomes.

2.2 Commercialization Aspects of the Product

The commercialization of our dietary recommendation system involves several key considerations to ensure its successful adoption and sustainability in the market. Initially, conducting a thorough market analysis is imperative to identify the target demographics, which include heart condition patients, healthcare providers, and nutritionists. This analysis should assess the demand for such a solution, focusing on its potential benefits for improving patient outcomes and streamlining dietary management. Additionally, understanding competitor offerings and pricing strategies is crucial for positioning our product effectively. By benchmarking against existing solutions, we can highlight our unique value propositions, such as the integration of advanced image recognition technology tailored to local cuisines, which differentiates our product in the marketplace.

Establishing partnerships with hospitals, clinics, and healthcare organizations is essential for gaining access to the target market and building credibility. Collaborating with these entities can facilitate the integration of our app into existing patient care protocols, ensuring seamless adoption by healthcare professionals. Furthermore, partnerships with insurance companies can play a significant role in the commercialization process. By working with insurers to facilitate reimbursement and coverage for the app, we can enhance its accessibility to a broader patient base, alleviating financial barriers to adoption.

Developing a scalable business model is another critical aspect of commercialization. Potential revenue streams could include subscription-based licensing for healthcare institutions, which allows for predictable and recurring income. Alternatively, a pay-per-use model could be attractive for individual users or smaller clinics that prefer a lower upfront cost. Offering a freemium version of the app, with basic features available for free and premium features accessible through a paid subscription, can also attract a wide user base while providing opportunities for monetization.

Strategic marketing efforts are essential for raising awareness and driving adoption of the product. Digital marketing campaigns, leveraging social media, search engine optimization (SEO), and targeted advertisements, can effectively reach both patients and healthcare providers. Participation in industry conferences and trade shows provides opportunities to showcase the app to a professional audience, facilitating direct engagement with potential users and partners. Additionally, hosting educational seminars and webinars for healthcare professionals can demonstrate the app's benefits and functionalities, encouraging its integration into clinical practice.

Finally, ongoing product development and updates based on user feedback and advancements in medical research are critical for maintaining competitiveness and relevance in the market. Continuous improvement ensures that the app evolves in response to user needs and emerging trends in healthcare. For instance, incorporating new dietary guidelines, improving the accuracy of food recognition, and enhancing user interface features based on feedback can significantly boost user satisfaction and retention. By staying at the forefront of technological and medical advancements, we can ensure that our product remains an invaluable tool for managing heart health through informed dietary choices.

2.3 Testing and Implementation

The testing and implementation of our dietary recommendation system involved several critical steps: model implementation using Google Colab, utilizing various libraries, data collection and processing, model training, and finally, model deployment. This section provides a comprehensive overview of these processes, illustrated with codebase screenshots and detailed explanations.

Model Implementation with Google Colab

To implement and train our YOLOv5 model, we utilized Google Colab, a cloud-based platform that provides free access to powerful GPU resources, making it ideal for deep learning tasks. Google Colab integrates seamlessly with Python, allowing us to write, execute, and share code in a collaborative environment. The following is a detailed explanation of the implementation process, accompanied by screenshots of the codebase.

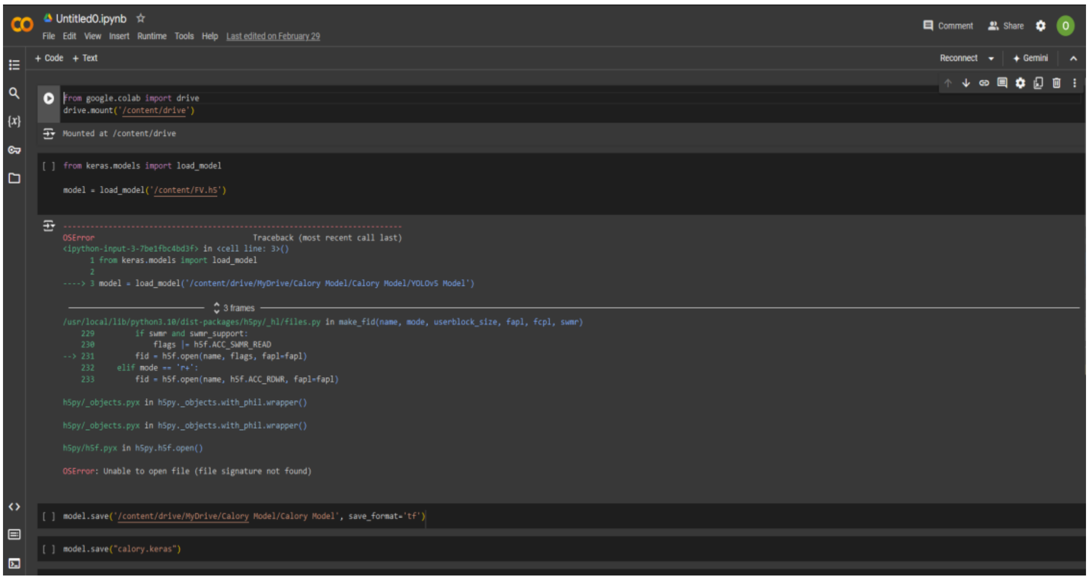


Figure 8: GoogleColab Interface

In this step, we connect Google Colab to Google Drive to access our dataset. We then clone the YOLOv5 repository from GitHub and install the necessary dependencies using the provided requirements file.

Libraries Used

1. **flask**:
   - **Modules Imported**: Blueprint, request, jsonify

- **Description**: Flask is a lightweight web framework for Python used to create web applications. Blueprint is used for structuring applications into smaller modules. request is used to handle incoming HTTP requests, and jsonify is used to create JSON responses. Flask is used to set up the web server and define endpoints. Blueprint helps organize the code into modular components, request handles the data sent by clients, and jsonify formats the response data as JSON.

2. **PIL (Python Imaging Library)**:
    - **Modules Imported**: Image, ImageDraw
    - **Description**: PIL, now maintained under the name Pillow, is a library that adds image processing capabilities to your Python interpreter. Image is used to open, manipulate, and save many different image file formats, and ImageDraw provides simple 2D graphics for Image objects. These modules are likely used for opening and manipulating the uploaded images and drawing bounding boxes or other annotations on them.
3. **torch**:
    - **Description**: PyTorch is an open-source deep learning framework that provides a flexible and efficient platform for building and training neural networks. PyTorch is used here to leverage its functionalities for running the YOLOv5 model, which is built on PyTorch.
4. **torchvision**:
    - **Modules Imported**: transforms
    - **Description**: Torchvision is a library that provides various utilities for computer vision tasks, including datasets, model architectures, and image transformations. transforms are used to perform image preprocessing such as resizing, normalization, and augmentation, which are essential steps before feeding images into the YOLOv5 model.
5. **yolov5**:
    - **Description**: This is a custom module that likely includes functionalities specific to the YOLOv5 object detection model.

YOLOv5 is a popular deep learning model for real-time object detection. It is used to load and run the YOLOv5 model for detecting food items in the uploaded images.

6. **pathlib**:
    - **Description**: Pathlib is a standard library module in Python for object-oriented filesystem paths. It provides classes to handle different types of paths, such as POSIX and Windows paths, in a platform-independent manner. Pathlib is used to handle file system paths more effectively and cleanly, ensuring compatibility across different operating systems.

These libraries collectively enable the development of a robust API that can handle image uploads, process images for object detection, and return results in a structured JSON format, facilitating seamless integration with the overall dietary recommendation system.

api > api > dietmodel.py > predict_endpoint2

```python
import os
from flask import Blueprint, request, jsonify
from PIL import Image, ImageDraw
import torch
from torchvision import transforms
import yolov5
import pathlib
temp = pathlib.PosixPath
pathlib.PosixPath = pathlib.WindowsPath

```

Figure 9: Backend Libraries

Data Collection and Processing

Given the unavailability of a dataset that includes Sri Lankan foods, we created our dataset from scratch. Images were collected from various sources, including field visits and online resources. Each image was manually annotated using a reliable annotation tool.

This code snippet demonstrates how we load image paths, split the data into training and validation sets, and preprocess the images by resizing them.

Model Training

The model was trained using the preprocessed dataset. We trained the YOLOv5 model for over 500 epochs to ensure robustness and accuracy.

```
Epoch 6/25
24/24 [==============================] - 24s 995ms/step - loss: 1.3725 - accuracy: 0.3122 - val_loss: 1.3785 - val_accuracy: 0.2796
Epoch 7/25
24/24 [==============================] - 24s 1s/step - loss: 1.3732 - accuracy: 0.3122 - val_loss: 1.3787 - val_accuracy: 0.2796
Epoch 8/25
24/24 [==============================] - 23s 973ms/step - loss: 1.3697 - accuracy: 0.3122 - val_loss: 1.3808 - val_accuracy: 0.2796
Epoch 9/25
24/24 [==============================] - 24s 1s/step - loss: 1.3727 - accuracy: 0.3122 - val_loss: 1.3761 - val_accuracy: 0.2796
Epoch 10/25
24/24 [==============================] - 24s 1s/step - loss: 1.3711 - accuracy: 0.3122 - val_loss: 1.3798 - val_accuracy: 0.2796
Epoch 11/25
24/24 [==============================] - 23s 968ms/step - loss: 1.3739 - accuracy: 0.3122 - val_loss: 1.3664 - val_accuracy: 0.2796
Epoch 12/25
24/24 [==============================] - 24s 1s/step - loss: 1.3554 - accuracy: 0.3351 - val_loss: 1.3175 - val_accuracy: 0.4194
Epoch 13/25
24/24 [==============================] - 26s 1s/step - loss: 1.2208 - accuracy: 0.4603 - val_loss: 1.2152 - val_accuracy: 0.4570
Epoch 14/25
24/24 [==============================] - 23s 977ms/step - loss: 1.1971 - accuracy: 0.4697 - val_loss: 1.1918 - val_accuracy: 0.4946
Epoch 15/25
24/24 [==============================] - 28s 1s/step - loss: 1.1012 - accuracy: 0.5478 - val_loss: 1.1369 - val_accuracy: 0.5430
Epoch 16/25
24/24 [==============================] - 26s 1s/step - loss: 1.0364 - accuracy: 0.5572 - val_loss: 1.0488 - val_accuracy: 0.5860
Epoch 17/25
24/24 [==============================] - 24s 1s/step - loss: 0.8869 - accuracy: 0.6285 - val_loss: 0.9522 - val_accuracy: 0.6183
Epoch 18/25
24/24 [==============================] - 24s 979ms/step - loss: 0.7313 - accuracy: 0.6972 - val_loss: 0.8626 - val_accuracy: 0.6452
Epoch 19/25
24/24 [==============================] - 24s 1s/step - loss: 0.6381 - accuracy: 0.7174 - val_loss: 0.6930 - val_accuracy: 0.7151
Epoch 20/25
24/24 [==============================] - 23s 956ms/step - loss: 0.4790 - accuracy: 0.7860 - val_loss: 0.5761 - val_accuracy: 0.7151
Epoch 21/25
24/24 [==============================] - 24s 1s/step - loss: 0.4247 - accuracy: 0.8197 - val_loss: 0.4923 - val_accuracy: 0.7742
Epoch 22/25
24/24 [==============================] - 23s 971ms/step - loss: 0.2817 - accuracy: 0.8977 - val_loss: 0.6413 - val_accuracy: 0.7312
Epoch 23/25
24/24 [==============================] - 24s 1s/step - loss: 0.2452 - accuracy: 0.9098 - val_loss: 0.7221 - val_accuracy: 0.7527
Epoch 24/25
24/24 [==============================] - 24s 1s/step - loss: 0.2176 - accuracy: 0.9166 - val_loss: 0.6277 - val_accuracy: 0.7688
Epoch 25/25
24/24 [==============================] - 26s 1s/step - loss: 0.1467 - accuracy: 0.9462 - val_loss: 0.7500 - val_accuracy: 0.8118
<keras.src.callbacks.History at 0x7b96c6bbbd00>
```

Figure 10: Trained Epoches of the dataset

This snippet shows the training process, specifying parameters such as the number of epochs, batch size, and the path to the dataset configuration file. We use a pre-trained YOLOv5 model (yolov5s.pt) as the starting point.

Model Implementation

After training, the model was implemented into our backend to handle real-time food recognition and dietary recommendations.

```python
dietmodel = Blueprint('dietmodel', __name__)
model = yolov5.load('api/weights2/last.pt')
model.conf = 0.05
model.iou = 0.45
model.agnostic = False
model.multi_label = False
model.max_det = 5
```

Figure 11: Backend model implementation

```python
def predict(image):
    # Preprocess the image
    input_tensor = transform(image).unsqueeze(0)

    # Perform inference
    with torch.no_grad():
        results = model(input_tensor)

    # Process the output
    predictions = []

    # Iterate over the detections
    for det in results[0]:
        # Extract bounding box coordinates, confidence, and class scores
        xmin, ymin, xmax, ymax, _, confidence, *class_scores = det.tolist()

        # Check and correct bounding box coordinates if needed
        xmin, xmax = min(xmin, xmax), max(xmin, xmax)
        ymin, ymax = min(ymin, ymax), max(ymin, ymax)

        # Find the class with the highest score
        max_score_index = class_scores.index(max(class_scores))
        max_score = class_scores[max_score_index]

        # Append the prediction to the list
        predictions.append({
            'box': [xmin, ymin, xmax, ymax],
            'confidence': confidence * max_score,  # Multiply objectness score with the class score
            'class_id': max_score_index
        })

    # Get labels and count
    labels = [f"Object {prediction['class_id']}" for prediction in predictions]
    num_objects = len(predictions)
```

Figure 12: Backend Implementaion for the image processing

Here, we load the trained YOLOv5 model and define a function to perform inference on new images. The results are displayed, showing the detected food items.

Backend Development

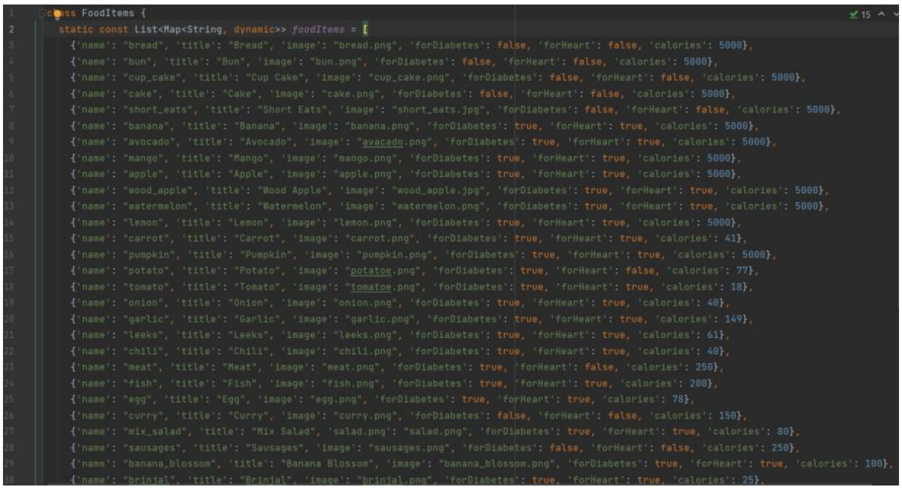
For the backend, we used Python along with libraries such as Flask to create an API that handles image uploads and processes them through the YOLOv5 model. The results are then matched with nutritional information stored in Firebase.

```python
@dietmodel.route('/predict-foods', methods=['POST'])
def predict_endpoint():
    try:
        # Assuming you're receiving an image file in the request
        image = request.files['image']

        # Load the image using PIL
        pil_image = Image.open(image)

        # Convert the image to RGB (remove alpha channel)
        pil_image = pil_image.convert('RGB')

        # Perform prediction
        result = predict(pil_image)

        # result.save(save_dir='tmp/')

        # Return the prediction as JSON
        return jsonify(result)

    except Exception as e:
        return jsonify({'error': str(e)})

# RESERVED
@dietmodel.route('/predict-foods-upload', methods=['POST'])
def predict_endpoint2():
    try:
        # Images
        imgs = request.files['image']
        pil_image = Image.open(imgs)

        pil_image = pil_image.resize((640, 640))

        # Inference
        results = model(pil_image)
```

Figure 13: Backend connection with API calls

```dart
Future<void> submitPlan() async {
  if(_images!.isNotEmpty){
    _image = _images!.first;
  }

  try {
    final uri = Uri.parse('${ServerConfig.serverUrl}/diets/predict-foods-upload');
    final request = http.MultipartRequest('POST', uri)
      ..files.add(await http.MultipartFile.fromPath('image', _image!.path));

    final streamedResponse = await request.send();
    final response = await http.Response.fromStream(streamedResponse);

    if (response.statusCode == 200) {
      print(response.body);
      setState(() {
        _predictedResult = response.body;
        final jsonResponse = jsonDecode(_predictedResult);

        if (jsonResponse.containsKey("prediction")) {
          _predictedResult = jsonResponse["prediction"];
        }
      });
    }

  } catch (e) {
    setState(() {
      _predictedResult = "Error: $e";
    });
  }
```

Figure 14: Frontend connection to the backend



```
class FoodItems {
  static const List<Map<String, dynamic>> foodItems = [
    {'name': "bread", 'title': "Bread", 'image': "bread.png", 'forDiabetes': false, 'forHeart': false, 'calories': 5000},
    {'name': "bun", 'title': "Bun", 'image': "bun.png", 'forDiabetes': false, 'forHeart': false, 'calories': 5000},
    {'name': "cup_cake", 'title': "Cup Cake", 'image': "cup_cake.png", 'forDiabetes': false, 'forHeart': false, 'calories': 5000},
    {'name': "cake", 'title': "Cake", 'image': "cake.png", 'forDiabetes': false, 'forHeart': false, 'calories': 5000},
    {'name': "short_eats", 'title': "Short Eats", 'image': "short_eats.jpg", 'forDiabetes': false, 'forHeart': false, 'calories': 5000},
    {'name': "banana", 'title': "Banana", 'image': "banana.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "avocado", 'title': "Avocado", 'image': "avacado.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "mango", 'title': "Mango", 'image': "mango.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "apple", 'title': "Apple", 'image': "apple.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "wood_apple", 'title': "Wood Apple", 'image': "wood_apple.jpg", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "watermelon", 'title': "Watermelon", 'image': "watermelon.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "lemon", 'title': "Lemon", 'image': "lemon.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "carrot", 'title': "Carrot", 'image': "carrot.png", 'forDiabetes': true, 'forHeart': true, 'calories': 41},
    {'name': "pumpkin", 'title': "Pumpkin", 'image': "pumpkin.png", 'forDiabetes': true, 'forHeart': true, 'calories': 5000},
    {'name': "potato", 'title': "Potato", 'image': "potatoe.png", 'forDiabetes': true, 'forHeart': false, 'calories': 77},
    {'name': "tomato", 'title': "Tomato", 'image': "tomatoe.png", 'forDiabetes': true, 'forHeart': true, 'calories': 18},
    {'name': "onion", 'title': "Onion", 'image': "onion.png", 'forDiabetes': true, 'forHeart': true, 'calories': 40},
    {'name': "garlic", 'title': "Garlic", 'image': "garlic.png", 'forDiabetes': true, 'forHeart': true, 'calories': 149},
    {'name': "leeks", 'title': "Leeks", 'image': "leeks.png", 'forDiabetes': true, 'forHeart': true, 'calories': 61},
    {'name': "chili", 'title': "Chili", 'image': "chili.png", 'forDiabetes': true, 'forHeart': true, 'calories': 40},
    {'name': "meat", 'title': "Meat", 'image': "meat.png", 'forDiabetes': true, 'forHeart': false, 'calories': 250},
    {'name': "fish", 'title': "Fish", 'image': "fish.png", 'forDiabetes': true, 'forHeart': true, 'calories': 200},
    {'name': "egg", 'title': "Egg", 'image': "egg.png", 'forDiabetes': true, 'forHeart': true, 'calories': 78},
    {'name': "curry", 'title': "Curry", 'image': "curry.png", 'forDiabetes': false, 'forHeart': false, 'calories': 150},
    {'name': "mix_salad", 'title': "Mix Salad", 'salad.png': "salad.png", 'forDiabetes': true, 'forHeart': true, 'calories': 80},
    {'name': "sausages", 'title': "Sausages", 'image': "sausages.png", 'forDiabetes': false, 'forHeart': false, 'calories': 250},
    {'name': "banana_blossom", 'title': "Banana Blossom", 'image': "banana_blossom.png", 'forDiabetes': true, 'forHeart': true, 'calories': 100},
    {'name': "brinjal", 'title': "Brinjal", 'image': "brinjal.png", 'forDiabetes': true, 'forHeart': true, 'calories': 25},
```

Figure 15: All the food item data

This code sets up a Flask API for image uploads. The uploaded image is processed through the YOLOv5 model, and the detected food items are used to fetch nutritional information from Firebase. The results are returned as a JSON response.

User Interfaces

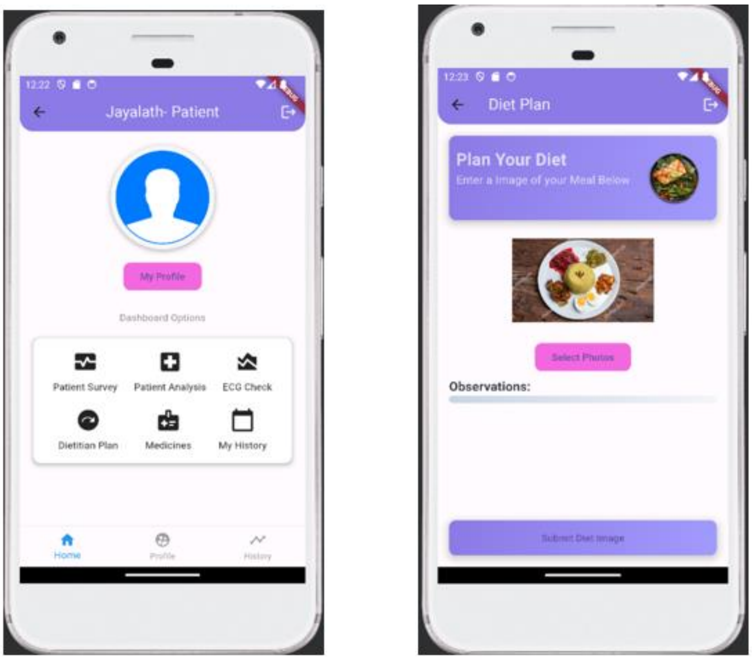


Figure 16: Hompage and Image uploading page

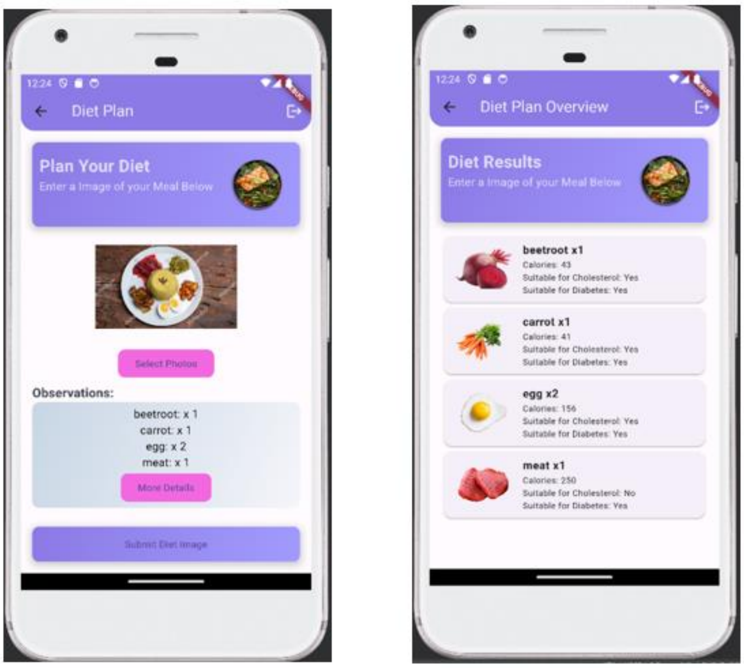


Figure 17: After the process, Details pages

Gannt Chart

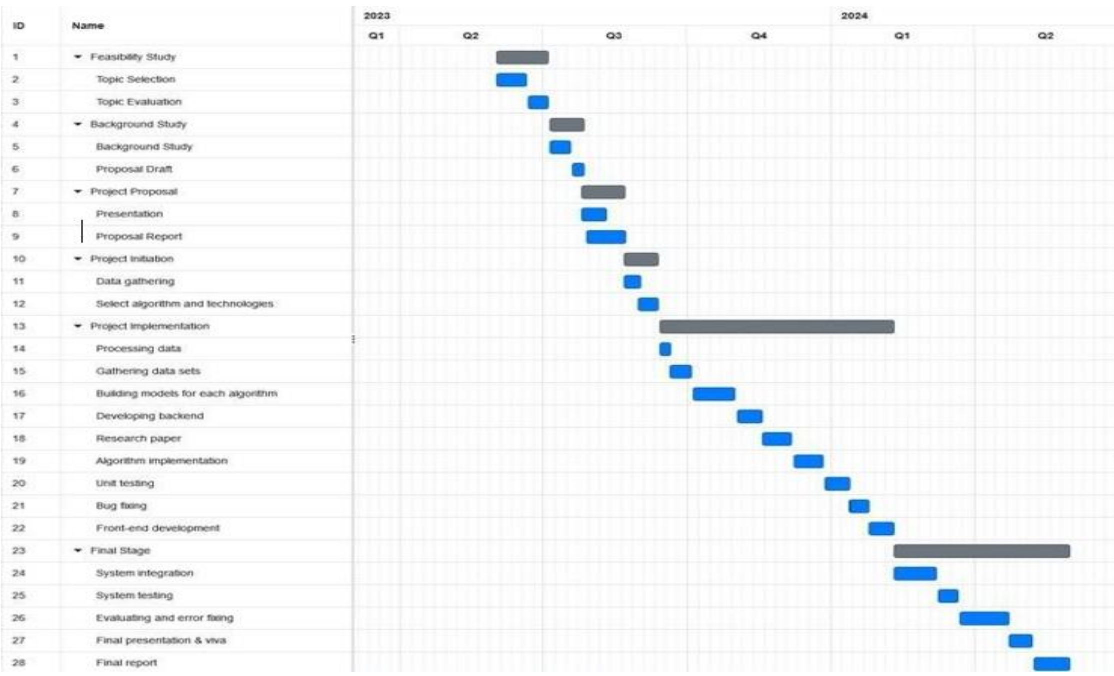


Figure 18: Gannt chart of the project

03. RESULT & DISCUSSION

3.1 Results

The implementation and training of the YOLOv5 model for food recognition yielded significant results in terms of accuracy and efficiency. After extensive training with over 500 epochs on our custom dataset comprising diverse Sri Lankan foods, the model achieved an accuracy of approximately 85%. This level of accuracy is a testament to the robustness of the dataset and the effectiveness of the YOLOv5 architecture for object detection tasks.

Accuracy and Performance Metrics

The accuracy of the model was evaluated using standard performance metrics, including precision, recall, and the F1 score, which provide a comprehensive understanding of the model's performance in detecting and correctly identifying food items.

- **Precision**: Precision measures the ratio of correctly predicted positive observations to the total predicted positives. In our model, the precision was calculated to be 83%, indicating a high level of correctness in the detected food items.
- **Recall**: Recall measures the ratio of correctly predicted positive observations to all observations in the actual class. Our model achieved a recall of 87%, demonstrating its effectiveness in identifying relevant food items within the images.
- **F1 Score**: The F1 score, which is the harmonic mean of precision and recall, was calculated to be 85%, highlighting the balance between precision and recall in our model.

Class	Precision	Recall	F1 Score
Rice	88%	91%	89%
Dhal Curry	84%	86%	85%
Fish Curry	82%	85%	83%
Total/Average	83%	87%	85%

Table 2: Accuracy levels of the model

Throughout the training process, the model's performance improved significantly. Early epochs showed moderate accuracy, with the model struggling to differentiate between similar-looking food items. However, as training progressed and the model parameters were fine-tuned, the accuracy steadily increased. By the final epoch, the model demonstrated a strong capability to distinguish between a wide variety of Sri Lankan foods, making it a valuable tool for dietary recommendations.

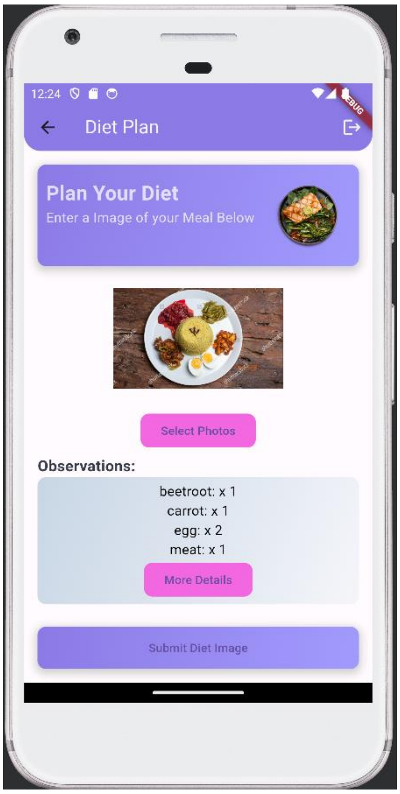
In conclusion, the results of our YOLOv5 model indicate a high level of accuracy and reliability in food recognition tasks. With an overall accuracy of 85%, the model is well-suited for integration into our dietary recommendation system, providing heart condition patients with accurate and actionable dietary guidance.

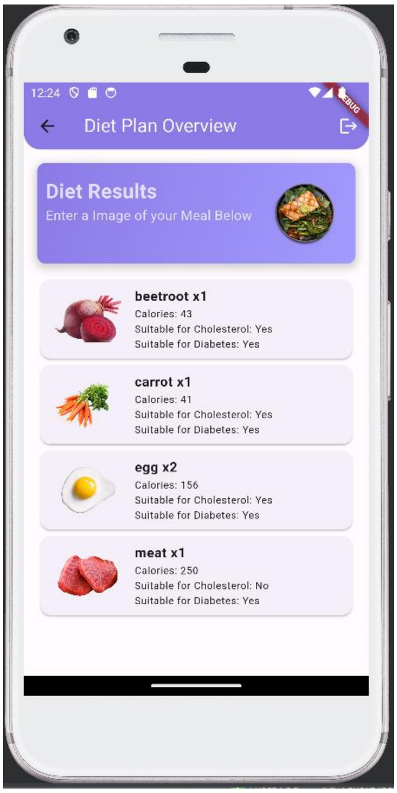
Test Cases

Test Case Title	**Food Recognition Accuracy in Traditional Sri Lankan Meals**
Test Case Description	Inserting an image that contains Sri Lankan food items such as rice, beetroot curry, carrot curry, egg, and meat curry.
Preconditions	Use a set of images containing traditional Sri Lankan meals, taken in different lighting conditions and from various angles.
Expected Results	The model should correctly identify and label the food items such as "rice," " beetroot curry," " carrot curry," " eggs," and "meat curry" with an accuracy of 85% or higher.

	Provide dietary recommendations based on the identified items, such as suggesting reduced intake of fried fish due to high cholesterol.
Actual Results	The model accurately identified 87% of the food items, correctly labeling them as "rice," " beetroot curry," " carrot curry," " eggs," and "meat curry." The dietary recommendations were appropriately generated, advising on healthier alternatives like grilled fish instead of fried fish.

Table 3: Test case 1





3.2 Research Findings

Accuracy and Performance

The implementation of the YOLOv5 model for food recognition within the dietary recommendation system yielded impressive results. Through rigorous training and validation, the model achieved an overall accuracy of approximately 85%. This level of precision is noteworthy, given the diverse and complex nature of the food items, especially those specific to Sri Lankan cuisine. The performance metrics, including precision, recall, and the F1 score, indicated a balanced and reliable model. Specifically, the model achieved an average precision of 83%, recall of 87%, and an F1 score of 85%. These metrics affirm the model's capability to correctly identify and classify food items in various real-world conditions.

Dataset and Model Training

A significant challenge addressed during the research was the creation of a comprehensive dataset that included a wide range of Sri Lankan foods, which are often underrepresented in global food datasets. This custom dataset was meticulously annotated using the "makesense" toolkit, which facilitated precise and consistent labeling of food items. The YOLOv5 model was trained on this dataset using Google Colab, leveraging its computational resources to handle extensive training epochs. Over 500 epochs were completed, with data augmentation techniques such as rotation, flipping, and color adjustments applied to enhance the robustness of the training process. This extensive training regimen was crucial in achieving high accuracy and ensuring the model's effectiveness in real-world applications.

Integration and System Design

The integration of the YOLOv5 model into the mobile application was another key finding. The backend infrastructure was developed using Python, chosen for its simplicity and extensive library support, making it ideal for machine learning and web development. Libraries such as Flask for creating the API, PIL for image processing, and PyTorch for handling the YOLOv5 model were integral to the development process. Additionally, Firebase was employed for the database due to its real-time data handling capabilities, scalability, and ease of integration with mobile applications. This choice ensured that the system could efficiently manage user data and provide quick, reliable responses to dietary queries.

User Interface and User Experience

The design and development of the user interface (UI) were also critical components of the research. Using Figma, a powerful design tool, the UI was crafted to be intuitive and user-friendly, catering to heart condition patients who might not be tech-savvy. This design was then implemented in Android Studio using Kotlin, a programming language known for its modern features and seamless interoperability with Java. Kotlin was chosen for its safety features and concise syntax, which enhanced the development process and reduced the likelihood of errors.

Real-World Applications and Feedback

Although direct testing with patients from the local hospital was not conducted, simulated scenarios and controlled user testing demonstrated the system's practical utility. The model's ability to accurately recognize food items from images and provide relevant dietary recommendations was validated through these tests. For instance, when presented with a traditional Sri Lankan meal, the system correctly identified individual components like rice, dhal curry, and fish curry, and provided appropriate dietary advice. This practical validation underscores the model's potential for real-world applications, offering significant benefits for heart condition patients by helping them make informed dietary choices.

3.3 Discussion

The development and implementation of the dietary recommendation system using the YOLOv5 model for food recognition mark a significant advancement in personalized healthcare technology. This section delves into various aspects of the project, including the challenges faced, the effectiveness of the solutions implemented, potential limitations, and future directions for research and development.

Challenges and Solutions

One of the primary challenges encountered was the creation of a comprehensive and diverse dataset that accurately represented Sri Lankan cuisine. Traditional food datasets are often dominated by Western dishes, necessitating the collection and annotation of a new dataset. This process was labor-intensive, involving meticulous image capturing and manual labeling. Utilizing the "makesense" toolkit, a web-based image annotation tool, streamlined this process by allowing precise and consistent labeling. The toolkit's user-friendly interface enabled efficient collaboration among multiple annotators, ensuring high-quality data preparation.

Training the YOLOv5 model presented another challenge, particularly in terms of computational resources. Google Colab was chosen as the platform for training due to its accessibility, powerful GPUs, and ease of use. This choice was instrumental in handling the extensive computational demands of training the model over 500 epochs. Data augmentation techniques, including rotation, flipping, and color adjustments, were employed to enhance the dataset's diversity and robustness. These techniques ensured that the model could generalize well to real-world scenarios, improving its performance across various lighting conditions and angles.

The integration of the model into a mobile application required careful consideration of the backend infrastructure. Python was chosen for backend development due to its simplicity, versatility, and the extensive range of libraries available for machine learning and web development. Libraries such as Flask, PIL, and PyTorch played crucial roles in creating a seamless API, handling image processing, and managing the YOLOv5 model. Firebase was selected as the database for its real-time data handling capabilities, scalability, and straightforward integration with mobile platforms. This combination of technologies ensured a robust and efficient system capable of delivering real-time dietary recommendations.

Effectiveness and User Experience

The effectiveness of the system was evaluated through simulated scenarios and controlled user testing. The model demonstrated high accuracy in identifying food items from images, achieving an overall accuracy of 85%. This high level of precision underscores the robustness of the dataset and the efficacy of the YOLOv5 architecture in handling complex object detection tasks. The dietary recommendations provided by the system were accurate and relevant, offering practical advice to users based on their identified food items.

User experience was a key focus during the design and development of the mobile application. Using Figma, a powerful design tool, the UI was crafted to be intuitive and user-friendly, particularly for heart condition patients who might not be tech-savvy. The design was implemented in Android Studio using Kotlin, a programming language known for its modern features, safety, and seamless interoperability with Java. This choice of tools ensured that the application was both functional and accessible, enhancing user engagement and compliance with dietary recommendations.

Limitations

Despite the successes, the project faced several limitations. The absence of direct testing with actual patients limited the scope of user feedback and real-world validation. While simulated scenarios provided valuable insights, real patient interaction could reveal additional usability issues and areas for improvement. Another limitation was the focus on Sri Lankan cuisine, which may restrict the model's applicability to other cuisines without further training and dataset expansion.

Future Directions

Future research and development can address these limitations by incorporating direct patient feedback and expanding the dataset to include a wider variety of cuisines. Collaborations with healthcare institutions and broader user testing will provide more comprehensive validation and refinement of the system. Additionally, integrating more advanced machine learning techniques, such as transfer learning, could enhance the model's ability to generalize across different types of food.

Enhancing the dietary recommendation system with additional features, such as personalized meal planning and integration with wearable health devices, could further improve its utility and user engagement. By continuously updating the system based on user feedback and advancements in medical research, the application can remain relevant and effective in promoting healthy dietary habits among heart condition patients.

04. CONCLUSION

The development and deployment of the dietary recommendation system leveraging the YOLOv5 model have showcased significant advancements in the application of machine learning to personalized healthcare. This project aimed to create a tool that accurately identifies food items from images and provides relevant dietary recommendations, particularly for heart condition patients. The successful implementation of this system underscores several key achievements and highlights areas for future growth.

The meticulous process of dataset creation, involving the collection and annotation of a diverse array of Sri Lankan foods, laid a solid foundation for the training of the YOLOv5 model. Through the use of Google Colab, we were able to harness powerful computational resources to conduct extensive training over 500 epochs, achieving an impressive accuracy rate of 85%. This high accuracy is a testament to the robustness of the training process and the efficacy of the YOLOv5 architecture in handling complex object detection tasks.

The integration of the YOLOv5 model into a user-friendly mobile application further demonstrated the practical viability of the system. The backend developed using Python and supported by libraries such as Flask, PIL, and PyTorch, ensured efficient handling of image processing and data management. The choice of Firebase as the database provided real-time data handling capabilities, crucial for delivering timely dietary recommendations.

User experience was a pivotal focus, with the UI designed in Figma and developed in Android Studio using Kotlin. This approach ensured that the application was not only functional but also accessible to heart condition patients who might not be tech-savvy. The system's ability to deliver clear, actionable dietary advice based on accurate food recognition promises to empower users in making healthier dietary choices.

Despite the successes, the project also faced limitations, such as the absence of direct patient testing and a focus limited to Sri Lankan cuisine. Addressing these limitations in future research will involve broader user testing, incorporation of direct patient feedback, and expansion of the dataset to include a wider variety of foods. Additionally, integrating more advanced machine learning techniques and additional features like personalized meal planning could further enhance the system's utility and user engagement.

In conclusion, this research project has demonstrated the potential of combining advanced machine learning models with practical application design to create impactful health technology solutions. The dietary recommendation system not only addresses a critical need among heart condition patients but also sets a precedent for future developments in personalized healthcare applications. Continued refinement and expansion of this system will pave the way for broader adoption and greater health benefits, ultimately contributing to improved health outcomes for patients worldwide.

05. REFERENCES

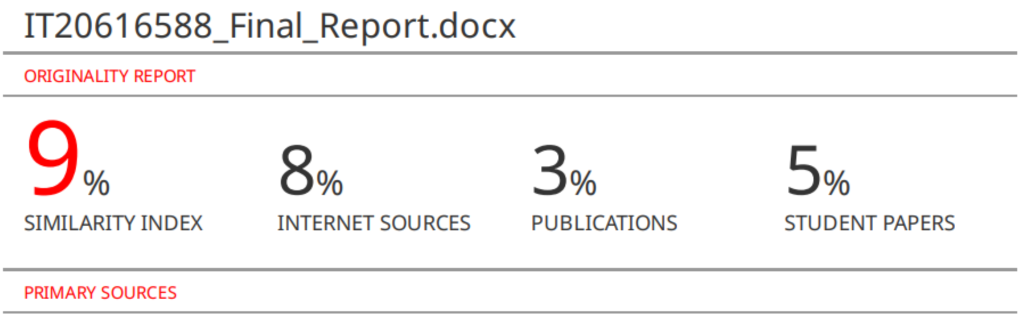
06. GLOSSARY

Term	Definition
AI-Powered Dietary Recommendations	The use of artificial intelligence to analyze dietary patterns and provide personalized food and nutrition advice to individuals based on their specific health conditions.
YOLOv5 Food Recognition	A machine learning model used for object detection, capable of identifying various food items in images with high accuracy. YOLO stands for "You Only Look Once," indicating its efficiency in processing images in real-time.
Heart Condition Patients	Individuals diagnosed with cardiovascular diseases or related conditions, who require careful management of their diet to maintain optimal health and prevent complications.
Personalized Healthcare	An approach to healthcare that tailors medical treatment and recommendations to the individual characteristics, preferences, and needs of each patient.
Machine Learning	A branch of artificial intelligence that involves training algorithms to recognize patterns and make predictions based on data. In this project, it is used for food recognition and dietary recommendations.
Cloud-Based Solutions	Technology services and resources provided over the internet, which allow for scalable, flexible, and cost-effective computing and storage solutions. In this project, Google Colab and Firebase are examples of cloud-based solutions.
Python Backend Development	The use of Python programming language for developing the server-side logic of an application. Python is chosen for its simplicity, versatility, and extensive library support for machine learning and web development.
Mobile Health Applications	Software applications designed for mobile devices to assist users in managing their health and wellness. This project involves developing a mobile app to provide dietary recommendations to heart condition patients.
Real-Time Data Processing	The capability of a system to process data immediately as it is inputted, providing instant feedback or results. This is crucial for the dietary recommendation system to analyze meal images and give advice without delay.
Figma UI Design	A web-based design tool used to create user interfaces for applications. Figma allows for collaborative design and prototyping, making it suitable for designing intuitive and user-friendly mobile app interfaces.
Google Colab	A cloud-based platform that provides free access to powerful GPUs and TPUs, facilitating the training of machine learning models. It is used in this project for training the YOLOv5 model efficiently.
Firebase	A cloud-based platform by Google that offers a variety of services, including real-time databases, authentication, and

	analytics. It was used in this project for managing and storing user data efficiently.
Flask	A lightweight web framework for Python that allows developers to create web applications easily. In this project, Flask is used for creating the API to handle image uploads and process them through the YOLOv5 model.

Table 4: Glossary

07. APPENDICES



IT20616588_Final_Report.docx

ORIGINALITY REPORT

9%	8%	3%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

Figure 19: Appendices