Acknowledgement

First of all we sincerely thank the Almighty GOD who is most beneficent and merciful for giving us the knowledge and courage to complete the project successfully.

We take this opportunity to express our sincere gratitude to all those who have helped us in completing this project successfully.

We wish to express our sincere gratitude to our Manager **Rev. Fr. Dr. Bobby Alex Mannamplackal** for providing infrastructural facilities for us, our Principal **Dr. Lillykutty Jacob** for the kind cooperation in all aspects of our project.

We owe a great depth of gratitude towards our Head of the Department **Dr. Juby Mathew** for his support during the entire course of this project work. We express our sincere and heartfelt gratitude to our Internal supervisor **Dr. Amal M R** and project coordinator **Mr.Syam Gopi** for the constant encouragement and support throughout the project. We also express our gratitude towards the supporting staff for their technical assistance.

We are indebted to our beloved teachers for their cooperation and suggestion throughout the project which helped us a lot. We also thank all our friends and classmates for their interest, dedication and encouragement shown towards the project. We convey our hearty thanks to our parents for their moral support, suggestion and encouragement to make this venture a success.

Abstract

The growing interest in food quality and safety requires the development of sensitive and reliable methods of analysis as well as technology for identifying freshness. FRUITEVAL endeavors to devise a method for evaluating the freshness and consumability of fruits by synergistically employing two distinct methodologies. The first method harnesses IoT sensors, encompassing temperature, humidity, and gas sensors, to continually assess the condition of the fruit in real-time. These sensors furnish crucial data points pivotal for gauging the fruit's freshness.

Concurrently, the project integrates various machine learning and deep learning models to analyse images of the fruit. By leveraging image recognition techniques, these models decipher visual cues indicative of the fruit's freshness and suitability for consumption. The integration of visual information thus provides another method of assessing freshness.

By combining IoT sensor data with advanced machine learning algorithms, the proposed methodology presents a holistic solution for discerning the freshness and consumability of fruits. The culmination of both methods enables the system to provide a comprehensive evaluation of each fruit's condition, thereby promoting food safety and mitigating food wastage.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Fruit quality assessment is a critical aspect of the agriculture and food industry, essential for ensuring consumer satisfaction and minimizing waste. Traditionally, this evaluation process has been performed through manual inspection, where inspectors assess the fruit's appearance, texture, and smell. However, these traditional methods are inherently subjective and can vary significantly between different inspectors, leading to inconsistencies. The advancements in technology, particularly in deep learning and the Internet of Things (IoT), present a significant opportunity to develop automated systems that can offer consistent, objective, and real-time assessments. These technologies can enhance the precision of fruit quality evaluation, streamline the process, and potentially reduce costs, making the supply chain more efficient and reliable. FRUITEVAL is designed to leverage these modern technologies to revolutionize the way fruit quality is assessed.

1.2 Motivation

The motivation for FRUITEVAL stems from the challenges faced in traditional fruit quality assessment methods and the opportunities presented by modern technology. Manual inspection methods are not only labor-intensive but also prone to human error and inconsistencies due to varying standards and levels of experience among inspectors. Additionally, there is a growing demand for real-time monitoring and analysis in the supply chain to quickly identify and address quality issues. Furthermore, while visual inspection is crucial, it often does not provide a complete picture of the fruit's quality. Chemical emissions, such as the concentration of alcohol in the air around the fruit, can offer additional insights into its ripeness and potential spoilage. By combining deep learning for visual assessment and IoT sensors for chemical analysis, FRUITEVAL aims to create a comprehensive and automated system for fruit quality evaluation.

1.3 Problem Definition

The traditional methods of fruit quality assessment are beset with several limitations. First, manual inspections are subjective and can vary between inspectors, leading to inconsistent quality evaluations. This subjectivity introduces a significant degree of error, which can affect the reliability of the assessment. Secondly, the process is inefficient; manually inspecting each fruit is time-consuming and labor-intensive, making it impractical for large-scale operations. Moreover, visual inspections alone do not provide a complete assessment of the fruit's quality. Indicators such as the concentration of alcohol emitted by the fruit, which can signal freshness or spoilage, are often not integrated into the evaluation process. There is a clear need for an automated system that can provide real-time, reliable data and reduce the reliance on manual inspections, thereby addressing these limitations. FRUITEVAL is developed to meet these needs and provide a robust solution.

1.4 Objective

The primary objective of FRUITEVAL is to develop an automated system for evaluating the quality of fruit using two complementary methods:

- Deep Learning Model: To predict the quality of the fruit from an image captured of it. The model will analyze features such as color, shape, and texture to assess the fruit's quality. Deep learning techniques, such as convolutional neural networks (CNNs), will be used for image classification and quality prediction.

- IoT Devices: To sense the concentration of alcohol in the air, which is an indicator of the fruit's freshness and potential spoilage. IoT sensors will be deployed to measure the alcohol concentration emitted by the fruit in real-time.

Both sets of data will be sent to a website where they will be displayed and analyzed. The website will provide a conclusive assessment of the fruit's quality and consumability based on the combined data from both methods. The specific objectives include achieving high accuracy in predicting fruit quality through image analysis and alcohol concentration measurement, providing real-time data and immediate feedback on fruit quality, developing a user-friendly website that displays the data in an intuitive and accessible manner for end-users, and integrating visual and chemical indicators for a holistic evaluation of fruit quality.

FRUITEVAL aims to deliver a reliable, efficient, and scalable solution for fruit quality assessment, benefiting producers by improving decision-making processes and ensuring consumers receive high-quality fruit.

Chapter 2

Software Requirements Specification

2.1 Introduction

2.1.1 Purpose

The purpose of the FRUITEVAL project is to develop an automated system for evaluating the quality and consumability of fruits using advanced technologies. This system aims to combine deep learning and IoT sensors to provide a comprehensive and reliable assessment of fruit quality. By analyzing both visual characteristics and chemical emissions, FRUITEVAL will offer a more accurate and objective evaluation compared to traditional manual inspection methods. The ultimate goal is to enhance the efficiency and consistency of fruit quality assessment, benefiting producers, distributors, and consumers by ensuring high-quality produce and reducing waste.

2.1.2 Scope

FRUITEVAL encompasses several key components and functionalities. The project involves the development and deployment of a deep learning model to analyze images of fruits, assessing features such as color, shape, and texture to predict quality. This requires collecting a dataset of fruit images, training a convolutional neural network (CNN), and integrating the model into the overall system.

In addition to image analysis, the project includes the integration of IoT devices to measure the concentration of alcohol in the air around the fruit. This data will offer insights into the fruit's freshness and potential spoilage. This involves selecting suitable sensors, setting up the necessary hardware, and ensuring accurate data collection and transmission.

A crucial aspect of the project is designing a data processing pipeline that aggregates and analyzes information from both the deep learning model and IoT sensors to provide a comprehensive assessment of fruit quality. The project also includes developing a user-friendly website that displays quality assessment results in an intuitive manner, featuring real-time data updates and visualization tools.

Ensuring real-time or near-real-time feedback on fruit quality involves optimizing the data processing and analysis pipeline. The project also includes rigorous testing and validation to ensure system accuracy, reliability, and robustness. FRUITEVAL is designed to be scalable, allowing for future enhancements such as additional sensor types or improved deep learning models.

2.1.3 Intended Use

This document provides a comprehensive overview of the FRUITEVAL application functionalities, outlining both formal and informal requirements to establish a clear context for the technical specifications.

2.1.4 Definitions and Acronyms

HTML (Hypertext Markup Language) is a widely-used markup language for creating and structuring web pages, defining the elements and their layout on a webpage. It provides a standardized format for organizing text, images, links, and other media, enabling the creation of interactive and visually appealing websites.

CSS (Cascading Style Sheets) is a language used to describe the presentation and visual styling of HTML documents. It allows the user to control the appearance of web pages, including layout, colors, fonts, and other visual elements, providing a consistent and attractive user experience.

JS (JavaScript) is a high-level, interpreted programming language primarily used for adding interactivity and dynamic behavior to web pages presentation of web components using JavaScript code.

HTTP(Hypertext Transfer Protocol): An application layer protocol for transmitting hypermedia documents, such as HTML, over a computer network.

CNN(Convolutional Neural Network) is a type of deep learning algorithm that is particularly well-suited for image recognition and processing tasks. It is made up of multiple layers, including convolutional layers, pooling layers, and fully connected layers. The architecture of CNNs is inspired by the visual processing in the human brain, and they are well-suited for capturing hierarchical patterns and spatial dependencies within images.

ResNet(Residual Neural Network) is a breakthrough deep learning model introduced in 2015 for image recognition. It tackles the challenge of training very deep neural networks by using residual learning, where layers learn residual functions with respect to their inputs. This approach enables ResNet to achieve high accuracy on complex tasks like the ImageNet competition. Its architecture includes identity skip connections, aiding in efficient training by maintaining gradient flow.

2.2 System Overview

2.2.1 System Description

The FRUITEVAL system automates fruit quality evaluation using advanced technologies. It incorporates a deep learning model for analyzing fruit images to predict quality based on color, shape, and texture. Additionally, IoT devices equipped with MQ3 sensors detect alcohol concentration around fruits, providing supplementary data on freshness and spoilage. Integrated into a unified system, these components enable real-time, comprehensive assessments of fruit quality via a dedicated web interface.

2.2.2 System Features

Key features of FRUITEVAL include:

Automated Fruit Quality Evaluation: Utilizes a deep learning model to analyze images of fruits captured by cameras. The model assesses various visual characteristics such as color, shape, and texture to predict the quality of the fruit.

IoT Integration with MQ3 Sensors: Integrates IoT devices equipped with MQ3 sensors. These sensors measure the concentration of alcohol in the air surrounding the fruits, providing additional data points related to freshness and potential spoilage.

Real-time Assessment: Enables real-time evaluation of fruit quality by processing data from both the deep learning model and MQ3 sensors. This allows for immediate feedback on the condition of the fruits.

User-Friendly Web Interface: Presents the results of the quality assessments through a dedicated web interface. Users can easily access and interpret the data, which includes detailed insights into the quality and consumability of the fruits.

Scalability and Flexibility Designed to be scalable, accommodating future enhancements and integrations. This includes the potential incorporation of additional sensors or improvements to the deep learning algorithms for enhanced accuracy.

2.2.3 Assumptions and Dependencies

The successful operation of FRUITEVAL relies on several key assumptions and dependencies. Firstly, the accuracy of the deep learning model in assessing fruit quality assumes a robust and diverse training dataset that effectively captures variations in fruit characteristics. The functionality of MQ3 sensors to accurately detect alcohol concentrations around fruits is essential for providing reliable data on freshness and spoilage. Stable and uninterrupted internet connectivity is critical for transmitting real-time data from IoT devices to the web interface, ensuring timely and accurate assessments.

Additionally, user proficiency in navigating and interpreting the web interface is assumed to facilitate effective use and understanding of the displayed fruit quality metrics.

Dependencies include the operational integrity of cameras and MQ3 sensors, which are pivotal for precise and consistent fruit quality evaluations. The availability and compatibility of deep learning frameworks for model training and deployment, along with web development frameworks for the user interface, are crucial software dependencies. Access to a comprehensive dataset of fruit images is necessary for training the deep learning model effectively. Regular maintenance and calibration of IoT devices are necessary to sustain their accuracy and reliability over time, ensuring consistent performance in fruit quality assessment.

These assumptions and dependencies collectively support the reliable and effective functioning of the FRUITEVAL system in automating fruit quality evaluation across diverse operational scenarios.

2.3 System Requirements

2.3.1 Functional Requirements

1. **User Interface:** The FRUITEVAL system will feature a user-friendly interface designed to facilitate intuitive interaction with the fruit quality analyzer. This interface will be accessible and straightforward, ensuring that users can easily navigate through the system. It will prominently display buttons that allow users to add images for analysis. These buttons will support both camera capture and file browsing methods, giving users flexibility in how they input images for assessment.

2. **Image Upload:** Users will have the capability to upload images of fruits through the system's interface. This functionality will be implemented using file selection, where users can choose images stored on their devices. Additionally, the system will integrate a webcam feature within the interface, enabling users to capture images directly. This dual approach to image upload ensures convenience and accessibility, accommodating different user preferences and technical setups.

3. **Image Analysis:** Upon uploading or capturing images, the system will perform thorough analysis to assess the quality of the fruits depicted. The analysis will evaluate various aspects such as color, size, texture, and defects present in the fruit images. Deep learning algorithms will be utilized extensively for this evaluation, ensuring that the assessments are accurate and reliable. These algorithms will leverage neural network models trained specifically for fruit quality assessment, enhancing the system's ability to provide insightful evaluations.

4. **Sensor Data Integration:** In addition to image analysis, the FRUITEVAL system will integrate data from IoT sensors to augment its fruit quality assessment capabilities. These sensors will measure parameters such as temperature, humidity, and potentially alcohol concentration in the air around the fruits. By incorporating this sensor data, the system will provide a more comprehensive analysis of fruit freshness and potential spoilage. This integration ensures that the assessments are not solely based on visual cues but also on real-time environmental data gathered through IoT devices.

5. **Result Display:** Following the analysis process, the system will present the results of the fruit quality assessment in a clear and understandable format to users. This presentation will include detailed insights derived from both image analysis and sensor data integration. Users will be able to view the final assessment of fruit freshness, supported by visual representations and textual summaries that convey the analysis findings effectively. The goal is to provide users with actionable information that aids in decision-making regarding fruit quality and consumption.

6. **Navigation and Redirection:** To enhance user experience, the FRUITEVAL system will employ effective navigation and redirection mechanisms within its interface. Based on user interactions, such as uploading images or viewing analysis results, the system will dynamically redirect users to appropriate pages or sections (e.g., upload, results). This seamless navigation ensures that users can easily progress through different stages of the assessment process without confusion. Furthermore, the system will maintain user selections and context across different pages, preserving continuity and allowing users to backtrack or explore additional information as needed.

2.3.2 Non Functional Requirements

1. **Performance:** The system strives to process image uploads and analyses efficiently, aiming for completion within a reasonable time frame, ideally under 5 seconds. This ensures timely feedback on fruit quality assessments for users. It is designed to handle multiple concurrent users adequately, maintaining acceptable performance levels during normal usage. However, under heavy loads, some degradation in performance may occur.

2. **Usability:** To cater to users with varying levels of technical expertise, the system emphasizes ease of use. The interface is intuitive, offering clear instructions and straightforward navigation at each step of the process. This approach enables users to upload images, interpret analysis results, and understand fruit quality assessments without unnecessary complexity.

3. **Reliability:** Users can expect consistent and generally accurate results from the system's fruit quality analysis across different conditions and user inputs. While the system aims for reliability, occasional variances may occur due to factors such as image quality or environmental changes. It includes error handling mechanisms to address issues gracefully, providing informative error messages when necessary to assist in troubleshooting and resolution.

4. **Security:** Ensuring the security and privacy of user-uploaded images is a top priority. The system employs encryption for data transmission and storage and implements stringent access controls to restrict data access to authorized personnel only. These measures mitigate risks of unauthorized access, data breaches, and ensure secure communication channels throughout system operations.

5. **Scalability:** The system is designed to be scalable, capable of handling increased workload and user demands as the user base grows. This scalability ensures that the system can expand its capacity and resources effectively to maintain performance levels under increasing usage. However, scaling efforts may require ongoing optimization and monitoring to manage resource allocation and maintain optimal performance.

6. **Compatibility:** The system aims for compatibility with major web browsers such as Chrome, Firefox, and Safari, ensuring seamless accessibility

across different browser platforms. It is also responsive and adaptable to various devices, including desktops, tablets, and smartphones. While efforts are made to optimize user experience across different devices, variations in performance and display may occur based on device specifications and browser capabilities.

2.4 External Dependencies

1. **TensorFlow:** TensorFlow is an open-source machine learning framework developed by Google. It provides tools for building and training deep learning models. In this project, TensorFlow is used extensively for defining the neural network architecture, compiling models with optimizers and loss functions, and training models on data.

2. **Keras:** Keras is an API designed to be user-friendly, modular, and extensible. It provides high-level abstractions for building and training deep learning models. In this project, Keras serves as the front end to TensorFlow, simplifying the process of defining layers, compiling models, and performing training operations.

3. **NumPy:** NumPy is a fundamental package for numerical computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays. In FRUITEVAL, NumPy is used for array manipulation, numerical operations on data, and handling image data arrays.

4. **Pandas:** Pandas is a powerful data analysis and manipulation library. It provides data structures like DataFrame, which allows for easy handling and analysis of structured data. In this project, Pandas is used for creating dataframes from file paths and labels, manipulating data, and preparing datasets for training.

5. **Matplotlib:** Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is used to generate plots, histograms, scatterplots, and other graphical representations of data. In FRUITEVAL, Matplotlib is utilized for visualizing images, accuracy and loss trends during model training, and generating confusion matrices.

6. **Seaborn:** Seaborn is a statistical data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics. In this project, Seaborn enhances the aesthetics of plots and visualizations, such as heatmap representations of confusion matrices.

7. **scikit-learn:** scikit-learn is a machine learning library for classical machine learning algorithms. It provides simple and efficient tools for data mining and data analysis. In FRUITEVAL, scikit-learn is used for metrics such as confusion matrix, classification report, and accuracy score evaluation. These metrics are essential for assessing the performance of the fruit quality classification model.

8. **InceptionResNetV2:** InceptionResNetV2 is a deep convolutional neural network architecture developed by Google. It is pre-trained on the ImageNet dataset and is commonly used for transfer learning tasks. In FRUITEVAL, InceptionResNetV2 serves as a feature extractor for images, leveraging its trained weights to extract meaningful features from fruit images without retraining the entire network.

9. **pathlib:** pathlib is a module in Python's standard library for handling filesystem paths as objects. It provides an object-oriented interface for working with filesystem paths and directory operations. In FRUITEVAL, pathlib is used for managing file paths to images and directories containing fruit quality data.

10. **os:** The os module provides functions for interacting with the operating system, such as navigating directories, manipulating paths, and handling file operations. It is used in FRUITEVAL for tasks like joining paths and listing directory contents.

11. **EarlyStopping:** EarlyStopping is a callback provided by TensorFlow/Keras that terminates model training when a monitored metric has stopped improving. It helps prevent overfitting by stopping training early when validation metrics no longer improve. In FRUITEVAL, EarlyStopping is used to monitor the validation loss and stop training if the loss does not decrease after a certain number of epochs.

12. **Arduino IDE:** The Arduino Integrated Development Environment (IDE) is an open-source software platform that enables the development and uploading of code to Arduino microcontroller boards. It provides a user-friendly interface, built-in libraries, and tools for writing, debugging, and compiling code in a simplified version of C++. The IDE supports various Arduino boards and is essential for programming microcontrollers used in IoT applications, such as integrating sensor data for the FRUITEVAL project.

13. **MQ3 Sensor:** The MQ3 gas sensor is designed to detect the presence of alcohol vapors in the air. It is commonly used in breathalyzers and other applications where alcohol detection is required. The sensor outputs an analog signal that varies with the concentration of alcohol, which can be read and processed by a microcontroller to determine the air quality.

14. **ESP32 Board:** The ESP32 is a powerful microcontroller with integrated Wi-Fi and Bluetooth capabilities, widely used in IoT projects. It supports various sensors and modules, including the MQ3 sensor, and provides the necessary interfaces and processing power to handle real-time data acquisition and processing. The ESP32 board is programmed using the Arduino IDE, making it a versatile and essential component of the FRUITEVAL project.

2.5 Use Case Diagram

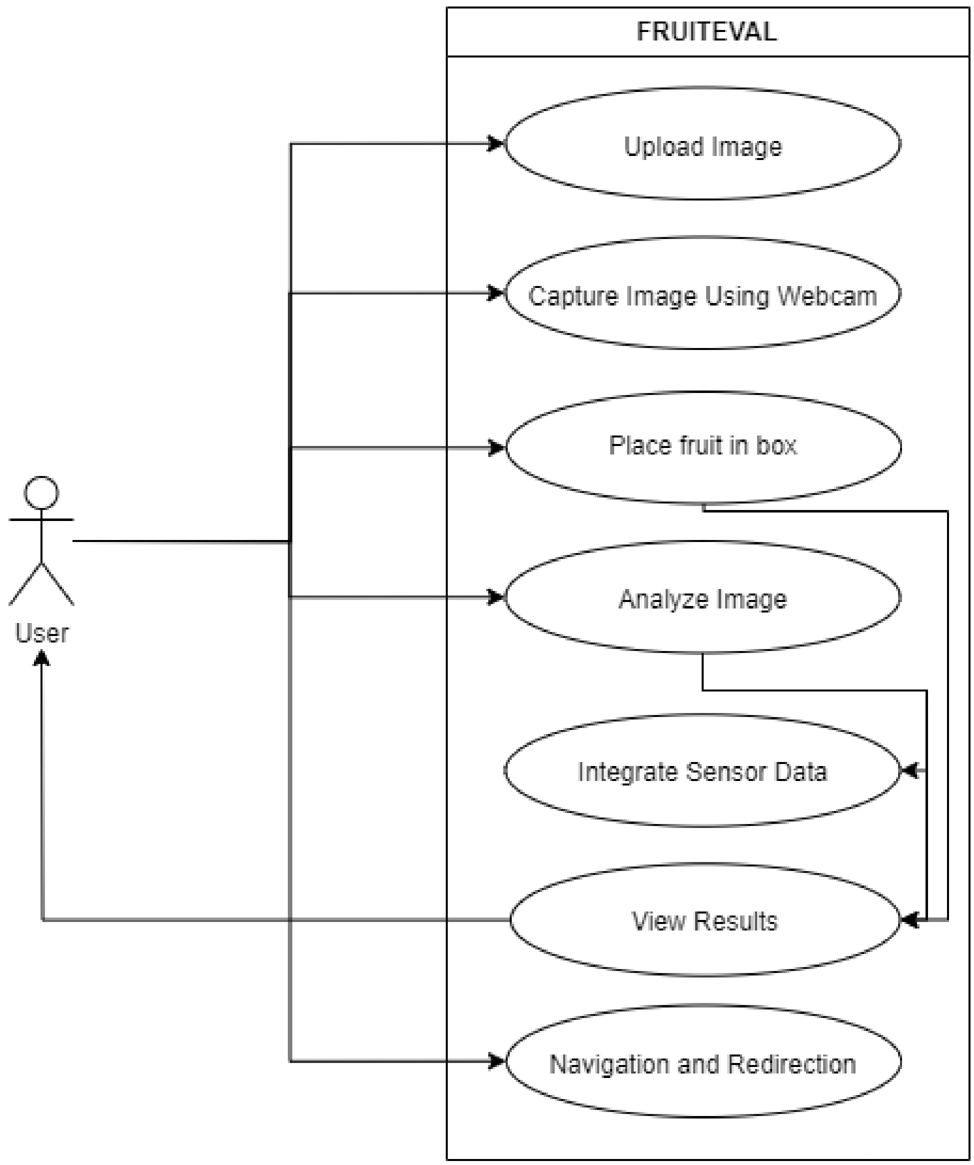

Figure 2.1: Use Case diagram showing the interaction of user with FRUITEVAL

2.6 Activity Diagram

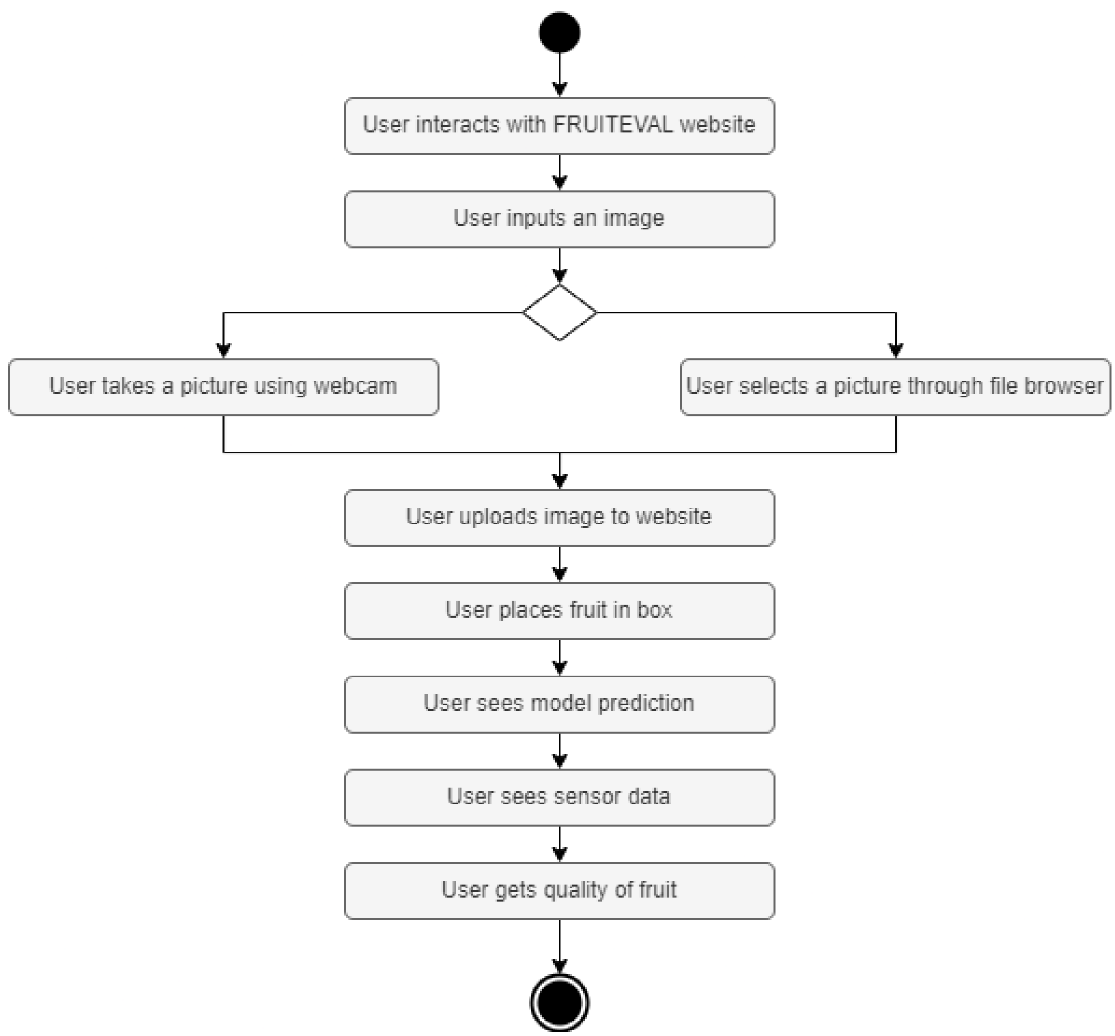

Figure 2.2: Activity diagram of FRUITEVAL

Chapter 3

Software Adopted

The FruitEval system is designed to provide an efficient and accurate method for assessing the quality of fruits using a combination of IoT sensors and camera modules. Leveraging modern web technologies, the system creates an interactive and user-friendly interface. The following sections provide an overview of the various components of the FruitEval system, detailing how each technology contributes to the system's functionality and user experience.

3.1 HTML

HTML (HyperText Markup Language) is used to structure the content of the FruitEval web application. The primary components of the HTML structure include:

- **Main Page (FruitIndex.html):** This page contains the user interface elements for initiating the fruit quality analysis, including buttons for adding images either through a camera or by browsing files. It also includes modal dialogs for file selection and camera usage.
- **Upload Page (upload.html):** This page handles the display of the selected option (camera or file upload) and manages the actual file input for analysis.
- **Output Page (output.html):** This page displays the results of the fruit quality analysis, including sensor data and the final assessment of fruit freshness.

3.2 CSS

CSS (Cascading Style Sheets) is used to style the HTML elements, making the user interface visually appealing and responsive. The CSS styles for the FruitEval system include:

- **Preloader Styling:** Ensures a smooth loading animation while the application is initializing.
- **Button Styles:** Defines the appearance and hover effects for various buttons used throughout the application (e.g., "Add Image", "Use Camera", "Browse from Files").
- **Modal Styling:** Styles the modal dialogs that appear for file selection and camera usage, including their visibility, layout, and transition effects.
- **Camera Section Styling:** Provides styling for the video feed when using the camera to capture images, ensuring it fits well within the layout of the application.

3.3 JavaScript

JavaScript (JS) is used to add interactivity to the FruitEval system. It handles user interactions, controls the camera functionality, and manages file uploads. Key JavaScript functionalities include:

- **Preloader Management:** : Ensures the preloader is hidden once the content is fully loaded.
- **Modal Dialogs:** Scripts manage the opening and closing of modal dialogs, allowing users to select between using the camera or browsing files.
- **Camera Access:** Utilizes the WebRTC API to access the user's webcam, enabling live video feed and image capture functionality.

- **File Upload Handling:** Manages file input events, allowing users to browse and select image files for analysis. This includes updating the display with the selected file's name and triggering the analysis process.
- **Redirection and State Management:** JavaScript handles the redirection between different pages (main, upload, output) and maintains the state of the selected options (camera or file upload).

3.4 Jupyter Notebook

Jupyter Notebook is an open-source web application that allows you to create and share documents containing live code, equations, visualizations, and narrative text. It provides an interactive environment for data analysis, scientific computing, and prototyping. Key features include:

1. **Notebook Format:**
    - Jupyter Notebook uses a notebook file format (.ipynb) that combines code, text, and multimedia elements.
    - Notebooks consist of a series of cells, each of which can contain code (executable cells) or text (markdown cells).
2. **Interactive Computing:**
    - Jupyter Notebook supports interactive computing in various programming languages, including Python, R, Julia, and more.
    - You can execute code cells individually or run the notebook sequentially.
3. **Rich Text Support:**
    - Jupyter Notebook supports the use of Markdown, allowing you to include formatted text, headings, links, images, and mathematical equations using LaTeX syntax.
    - Markdown cells can be used for documentation, explanations, and adding narrative context to your analysis.
4. **Code Execution and Output:**

- Jupyter Notebook provides an environment for executing code cells, which can include programming code in the supported languages.
- The output of code cells is displayed directly below the cell, making it easy to view and analyze results.

5. **Data Visualization:**
   - Jupyter Notebook integrates with popular data visualization libraries like Matplotlib, Seaborn, and Plotly, allowing you to generate interactive charts, graphs, and plots directly in the notebook.
   - Visualizations can be displayed inline within the notebook or in separate interactive windows.

3.5 Deep Learning Framework

We propose a Convolutional Neural Network model for the classification task. The current section defines the fundamental concepts of the model and the parameters of the proposed model.

3.5.1 Convolutional Neural Networks (CNNs)

CNNs are a class of neural networks designed explicitly for image-processing tasks. CNNs use convolutional and pooling layers to extract features from an input image.

Convolutional layers work by convolving a set of learnable filters (kernels) over the input image to produce feature map. The filters are designed to detect specific patterns in the image, such as edges or corners. Pooling layers are used to downsample the feature maps produced by convolutional layers, reducing their size while retaining the most critical information. The most common type of pooling layer is max pooling, which takes the maximum value from each subregion of the feature map.

CNNs have succeeded highly in image classification tasks, achieving State-of-the-Art performance on benchmark datasets such as ImageNet. However, they are limited in their ability to capture global relationships between different parts of an image.

3.5.2 Residual Neural Network

A residual neural network (also referred to as a residual network or ResNet) is a seminal deep learning model in which the weight layers learn residual functions with reference to the layer inputs. ResNet behaves like a highway network whose gates are opened through strongly positive bias weights.

Traditional deep neural networks often suffer from the degradation problem, where increasing the depth of the network leads to higher training error rates. This is counterintuitive because, theoretically, a deeper network should model more complex functions and thus reduce the training error. However, in practice, as the network depth increases, the network starts to perform worse on the training data, even when the vanishing gradient problem is mitigated.

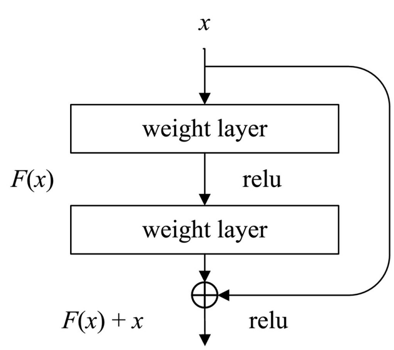

Figure 3.1: Residual learning: a building block

ResNet addresses the degradation problem by introducing a residual learning framework. Instead of trying to learn the desired underlying mapping directly, ResNet learns the residual mapping.Formally, denoting the desired underlying mapping as $\mathcal{H}$, we let the stacked nonlinear layers fit another mapping of $\mathcal{F}(x) := \mathcal{H}(x) - x$. The original mapping is recast into $\mathcal{F}(x) + x$.

The formulation of $\mathcal{F}(x) + x$. can be realized by feedforward neural networks with “shortcut connections” (Figure 3.1). Shortcut connections are those skipping one or more layers. The shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers (Figure 3.1). These connections skip one or more layers and add the input directly to the output of a stack of layers. This can be represented as

$y = \mathcal{F}(x, \{W_i\}) + x$, where $\mathcal{F}(x, \{W_i\})$ represents the residual function (a series of layers with weights $\{W_i\}$) and § is the input that is added directly to the output of $\mathcal{F}$.

3.5.3 Inception-ResNet-v2

Inception-ResNet-v2 is a convolutional neural architecture that incorporates residual connections to improve its performance. This architecture is based on the Inception family of architectures but enhances it by adding residual connections in place of the filter concatenation stage of the Inception architecture.

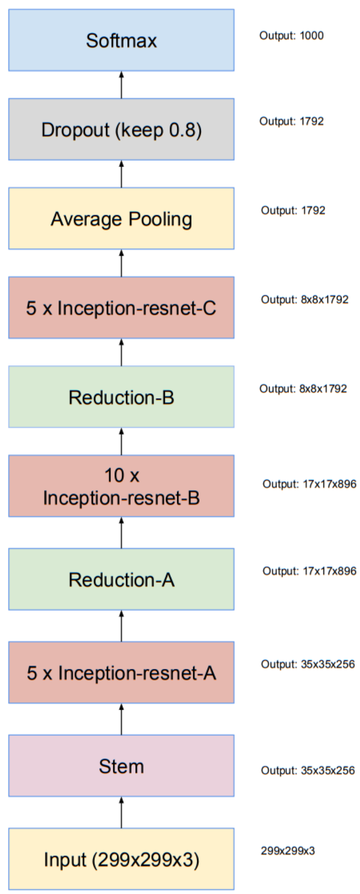

Figure 3.2: Overall architecture of Inception-ResNet-v2

The general architecture of Inception-ResNet-v2 includes:

Stem: The model starts with an input image of size 299x299x3 (height, width, channels). The initial layers consist of several convolutions and max-pooling operations to reduce the spatial dimensions while increasing the depth of the feature maps.

Inception-ResNet Blocks

10 x Inception-ResNet-B: Each Inception-ResNet-B block combines multiple paths, including 1x1, 3x3, and 7x1 convolutions, followed by concatenation and an addition of the input (residual connection). ReLU activation is applied after the addition.

Reduction-B: Another reduction block that further decreases the spatial dimensions and increases the depth. Similar to Reduction-A, it uses various convolution layers and a max-pooling layer followed by concatenation.

5 x Inception-ResNet-C: This block is similar to Inception-ResNet-B but includes different convolution layers (e.g., 1x1, 3x3) and concatenation paths. A residual connection is added, followed by ReLU activation.

Final Layers: The final layers consist of average pooling, which reduces the spatial dimensions to 1x1 while maintaining the depth, and a dropout layer to prevent overfitting. The last layer is a fully connected layer with a softmax activation function, which outputs class probabilities.

Chapter 6

Conclusion

This mini-project has successfully integrated advanced technologies to evaluate the quality and consumability of fruits. By combining these methodologies and seamlessly integrating them into a web-based platform, we gained valuable insights into effective fruit quality assessment.

The deep learning model, utilizing transfer learning with InceptionResNetV2, played a pivotal role in categorizing fruits into good and bad quality based on image analysis. Trained on a diverse dataset of fruit images, the model demonstrated robust performance, achieving high accuracy on the test set. Techniques such as data augmentation further enhanced its ability to generalize and make precise predictions, ensuring reliable assessments of fruit quality from visual inputs.

In conjunction with the deep learning model, an ESP32 microcontroller equipped with an MQ-3 alcohol sensor provided real-time data on alcohol concentration in the air surrounding the fruits. This parameter served as a critical indicator of fruit freshness, with a predefined threshold distinguishing between fruits safe for consumption and those not recommended.

Data outputs from both the deep learning model and the IoT device were seamlessly integrated into a user-friendly web interface. This interface not only presented predicted fruit quality based on image analysis but also conclusively determined their consumability status. This accessibility empowered users to make informed decisions regarding fruit quality and safety based on objective data.

In conclusion, this mini-project highlights the effectiveness of combining deep learning and IoT technologies for objective and efficient fruit quality evaluation. By leveraging these technologies, we've provided a practical solution to the complexities involved in fruit quality assessment, benefiting agricultural practices, retail standards, and consumer health. These findings contribute to ensuring the availability of high-quality and safe fruits in the market.

Looking ahead, future enhancements could focus on expanding the dataset to include a broader range of fruits and environmental conditions. Additionally, optimizing the deep learning model's parameters and integrating additional sensors could further enhance the accuracy and reliability of fruit quality assessments. Such advancements would extend the applicability of the system across diverse agricultural settings and enhance its overall utility.

Based on the insights gained, stakeholders in agriculture and technology sectors are encouraged to implement comprehensive quality control measures using AI-driven technologies, enhance public awareness about fruit quality and safety standards, and foster collaborative efforts to advance agricultural technology innovations. Ultimately, this mini-project contributes to advancing technology in agriculture, promoting healthier consumption habits, and supporting sustainable food practices, ensuring high-quality fruits are accessible and safe for all consumers.