Software Requirements Specification

for

Fruit Maturity Detection

Version 1.0 approved.

Prepared by

Saksham Rawat (209301285)

Manipal University Jaipur

8th February 2023

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The objective of this document is to analyze the requirements of the project "Fruit Maturity Detection". This system is designed with purpose of providing Inspection of Certain Fruits and assuring their quality. The major objective of this type of applications is to ensurea good product quality to the consumer.

1.2 Document Conventions

The document is typed in Times New Roman with font size 12 for general text, 16 for sub-headings and 20 for headings. All the headings and sub-headings are in bold, and the page headers are in italics.

1.3 Intended Audience and Reading Suggestions

This software requirements specification is intended for project developers using computer vision system using a digital camera and image analysis techniques. The remaining section of this specification describes functional requirements for the Fruit Maturity Detection.

1.4 Product Scope

The scope of the project is to make use of machine vision-based systems as they are automated, non-destructive, and cost- effective. This project is specially designed for the usage of the personnel working in the Agricultural sector. The system provides following features:

- Assessing the Maturity of a Fruit from images
- Information about ripeness of fruits from images.

1.5 References

- Raja Hamza and Mohamed Chtourou, "*Apple Ripeness Estimation using Artificial Neural Network*," in *International Conference on High Performance Computing & Simulation, 2005*

- Raymond Erz Saragih and Andi W.R. Emanuel, "*Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network*," in *3rd east Indonesia Conference on Computer and IT, 2021*

- Kaveri Kangune, Dr. Vrushali Kulkarni and Prof. Pranali Kosamkar, "*Grapes Ripeness Estimation using Convolutional Neural Network and Support Vector Machine*," in *Global Conference for Advancement in Technology (GCAT) Bangalore, India, 2019*

2. Overall Description

2.1 Product Perspective

Fruit Maturity Detection is a fully independent project aimed towards helping the Agricultural Community with the purpose of providing Inspection of Certain Fruits and assuring their quality.

- Determines ripeness of certain fruits from images.
- Assessing maturity of fruits from images.

2.2 Product Functions

The functions covered by the specifications include the following:

- Image Preprocessing of the Dataset.
- Extracting RGB from each image.
- Assessing ripeness of fruits from images.

2.3 User Classes and Characteristics

a) Admin:
- Responsible for managing the overall system and ensuring that all users have access to the necessary resources.
- Characteristics: Ability to add, delete and modify users, access to all information and functionality within the system, ability to generate reports.

b) Farmer:
- Responsible for uploading images of fruits to the system for analysis.
- Characteristics: Ability to upload images, access to results of analysis, ability to view historical results.

c) Agricultural Technologist:
- Responsible for analysing the images uploaded by the farmer and providing recommendations on fruit maturity.
- Characteristics: Ability to view images, access to algorithms and data analysis tools, ability to provide recommendations and results.

d) Agribusiness Manager:
- Responsible for monitoring the results of fruit maturity analysis and making decisions based on the results.
- Characteristics: Access to results of analysis, ability to view historical results, ability to make decisions based on results.

e) End User:
- The end user is the consumer of the fruit and is interested in the quality and maturity of the fruit.
- Characteristics: Ability to view results of analysis, access to information on fruit quality and maturity, ability to make informed purchasing decisions.

2.4 Operating Environment

- Hardware: The hardware required for this project would include a camera or a webcam and a computer.

- Software: The software required for this project includes image processing software OpenCV, machine learning libraries like scikit-learn, and the programming language Python.

- Database: A database will be used to store the images and data collected from the fruit maturity detection process. This could be a cloud-based database or a local database depending on the project requirements.

- User interface: A command-line interface (CLI) is used.

- Operating system: The operating system for this project will be a Windows based system.

- Environmental conditions: The operating environment for the fruit maturity detection system should be stable in terms of temperature and lighting conditions to ensure accurate results.

- Power supply: The project would require a reliable power supply to ensure that the system always remains operational.

2.5 Design and Implementation Constraints

I. Design Constraints:

- Accuracy: The system must accurately detect the maturity level of the fruit in real-time, without any false positives or false negatives.
- Cost: The system should be cost-effective and budget-friendly, avoiding the use of high-end and expensive hardware.
- Portability: The system must be portable, easy to install, and have a low maintenance cost.
- User-Friendliness: The system must have a user-friendly interface, making it easy for users with different levels of technical knowledge to operate it.
- Scalability: The system must be scalable to accommodate future growth, allowing for the integration of new features and technologies.

II. Implementation Constraints:

- Data Storage: The system must have a robust data storage mechanism, capable of storing large amounts of data, and making it easily accessible for analysis and reporting.
- Image Processing: The system must have an efficient image processing algorithm to accurately detect the maturity level of the fruit.

- Integration with Existing Systems: The system must be able to integrate with existing systems, such as ERP or CRM systems, to provide real-time data and insights.
- Security: The system must have a strong security mechanism to protect sensitive data and prevent unauthorized access.
- Technical Support: The system must have adequate technical support and maintenance services to ensure the smooth operation of the system.

2.6 User Documentation

- Introduction: Fruit maturity detection is a college project aimed at detecting the maturity of fruits using image processing techniques. The system uses a camera to capture images of the fruits and processes them to determine their level of maturity. This project is designed to help farmers, growers, and fruit packagers to automate the process of fruit maturity detection, saving time and effort.

- Requirements:

To run the fruit maturity detection system, the following requirements need to be met

i. A computer with a webcam or a standalone camera
ii. A Python 3.x installation
iii. OpenCV library installed.
iv. NumPy library installed.

- Setup and Installation:

i. Install Python 3.x on your computer if not already installed.
ii. Install the OpenCV library by running the following command in your terminal: pip install OpenCV-python.
iii. Install the NumPy library by running the following command in your terminal: pip install NumPy.
iv. Download the source code of the fruit maturity detection system from the repository.
v. Run the code using the following command in your terminal: python fruit_maturity_detection.py.

- Usage:

i. Connect your camera to your computer or place the fruits in front of your webcam.
ii. Run the fruit maturity detection system by using the above command.
iii. The system will capture the images of the fruits and process them to determine their level of maturity.
iv. The results will be displayed on the screen in the form of a percentage of maturity.

- Note:

The system has been trained on a limited number of fruits, and its accuracy may vary depending on the type and variety of fruit being tested. It is advised to perform additional testing and training on the system for improved results.

2.7 Assumptions and Dependencies

- Assumptions:

i. The fruit to be tested is a single species, for example, apples.
ii. The fruit is mature and ready for harvest.
iii. The fruit is placed on a flat surface for accurate measurement.
iv. The fruit is free from damage and not overripe.

- Dependencies:

i. A camera or image capturing device to take pictures of the fruit.
ii. A software or algorithm to analyse the images and determine the maturity level.
iii. A light source to provide consistent lighting for the image capture.
iv. Access to a database of fruit images to train the algorithm in recognizing maturity levels.
v. An internet connection for data transfer and access to online resources for research and development.
vi. A computer or device capable of running the image analysis software.

3. External Interface Requirements

3.1 User Interfaces

Welcome to the Fruit Maturity Detection Project!

Please choose one of the following options:

1. Scan a fruit.
2. View results
3. Exit

1

Please select the type of fruit you want to scan:

1. Apple
2. Banana
3. grapes

2

Please place the banana in the scanner for analysis.

Scanning...

Fruit Maturity Analysis:

- Fruit Type: Banana
- Colour: Yellow

- Ripeness: Ripe
- Recommendation: Ready to eat

1. Scan another fruit.
2. View results
3. Exit

3

Thank you for using the Fruit Maturity Detection Project. Have a great day!

3.2 Hardware Interfaces

- Camera: A high-resolution camera is used to capture images of the fruit and analyze its maturity level.
- LED Lights: LED lights are used to provide illumination for the camera and improve image quality.
- Display Screen: A display screen is used to present the results of the fruit maturity analysis in an easy-to-understand format.
- Input Devices: A keyboard and mouse are used to control the software and input data as needed.
- Storage: A storage device such as an SD card is used to save the data obtained from the camera and sensors.
- Power Supply: A power supply unit is used to provide the necessary power for all the hardware components.

3.3 Software Interfaces

- A software interface is a set of rules that govern how different software components or systems communicate with each other. It can be graphical, such as a user interface, or it can be a command-line interface (CLI) that uses text commands to interact with the software.
- A CLI is a text-based interface used to interact with a computer program or operating system. It provides a way to enter commands and receive responses from the system without using a graphical user interface. CLI is often used in software development, system administration, and networking tasks.
- In the context of fruit maturity detection using convolution neural networks, the software interface might allow users to input images of fruit and receive information about their maturity level. The CLI might allow users to specify parameters such as the type of fruit being analysed, the number of layers in the neural network, and the threshold for

determining maturity. Users might then receive output such as a numeric value indicating maturity level, or a visual display of the fruit with maturity levels indicated.

3.4 Communications Interfaces

- User Interface: A graphical user interface (GUI) is the most common communication interface for Fruit maturity Detection software. This interface enables the user to interact with the software by selecting the type of fruit, uploading an image of the fruit, setting parameters, and visualizing the output of the model. The GUI is designed to be intuitive, easy to navigate, and visually appealing.

- API: Fruit maturity Detection software using Convolution neural network can also be integrated with other systems through an application programming interface (API). The API allows other software to interact with the fruit maturity detection software using pre-defined commands, such as uploading an image and receiving the output of the model. The API can be designed to be platform-independent, making it easier to integrate with different systems.

- Command Line Interface: Some users prefer to interact with software using a command line interface (CLI) rather than a GUI. A CLI allows the user to enter commands in a terminal window, which are then executed by the software. This interface is typically used by advanced users who prefer a more efficient and streamlined way of interacting with the software.

- Web Interface: Fruit maturity Detection software can also be accessed through a web interface, which allows the user to interact with the software through a web browser. The web interface can be designed to be responsive, making it accessible from different devices, such as smartphones and tablets.

4. System Features

Fruit maturity detection software using Convolution Neural Network (CNN) is an advanced technology used in the agriculture industry for detecting the maturity level of fruits. The system features of this software are highly efficient, and they provide accurate results. This documentation discusses the system features of the fruit maturity detection software using CNN.

4.1 System Feature 1

Image Preprocessing: The first step in fruit maturity detection is image preprocessing. The software uses image preprocessing techniques such as image enhancement, color correction, and filtering to ensure that the input images are of high quality and free from noise. This process improves the accuracy of the CNN model in detecting the maturity level of fruits.

4.2 System Feature 2

Convolution Neural Network: The fruit maturity detection software uses a Convolution Neural Network (CNN) for image classification. CNN is a deep learning technique that is highly effective in detecting patterns and features in images. The system features a highly optimized CNN model that is trained using a large dataset of fruit images with known maturity levels. The CNN model is capable of accurately detecting the maturity level of fruits in real-time.

4.3 System Feature 3

User Interface: The software has a user-friendly interface that makes it easy to use. The user interface allows the user to select the type of fruit, input the image, and get the result. The interface also allows the user to adjust the threshold for the maturity level.

4.4 System Feature 4

Real-time Processing: The system features real-time processing, which allows the software to detect the maturity level of fruits in real-time. This feature is useful in situations where fruits need to be sorted based on their maturity level quickly.

4.5 System Feature 5

Accuracy: The fruit maturity detection software has a high level of accuracy in detecting the maturity level of fruits. The system has been trained using a large dataset of fruit images with known maturity levels, which ensures that the results are accurate.

4.6 System Feature 6

Compatibility: The software is compatible with various hardware platforms, including desktop computers, laptops, and mobile devices. This feature allows the software to be used in different environments and situations.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- Accuracy: The fruit maturity detection software should achieve a high level of accuracy in detecting the maturity of fruits. This means that the software should be able to correctly identify the maturity level of the fruit based on the features extracted by the CNN.

- Speed: The software should be able to perform fruit maturity detection quickly. This is particularly important in high-volume fruit processing environments where time is a critical factor.

- Robustness: The software should be able to handle different types of fruits and variations in their appearance due to factors such as lighting conditions, size, and shape. It should also be able to handle variations in the level of maturity of fruits.

- Scalability: The software should be scalable to handle large datasets of fruit images. It should also be able to handle multiple input sources and perform in real-time.

- User-friendly: The software should be easy to use and should have a user-friendly interface that allows users to interact with the system easily.

- Integration: The software should be able to integrate with other systems such as fruit sorting machines, packaging machines, and other hardware devices.

- Maintenance: The software should be easy to maintain, with regular updates and bug fixes, to ensure it remains accurate and reliable.

5.2 Safety Requirements

- Data quality and bias: The accuracy of the fruit maturity detection software depends on the quality of the data it is trained on. It is important to ensure that the data is representative of the different types of fruit being analyzed and is not biased towards certain types or stages of maturity. This can be achieved by collecting data from different sources and stages of production.

- Model testing and validation: Before deploying the software, it should undergo rigorous testing and validation to ensure its accuracy and reliability. This includes testing the software on a variety of fruit types and stages of maturity to ensure it can accurately detect maturity across different conditions.

- User safety: If the software is intended for use in a commercial setting, it is important to consider user safety. This may include developing protocols for safe operation and maintenance of the software, as well as providing adequate training to users.

- Cybersecurity: As with any software, it is important to ensure that the fruit maturity detection software is secure and protected from cyber threats. This may include implementing encryption and other security measures to protect data and prevent unauthorized access to the software.

- Regulatory compliance: Depending on the intended use of the software, it may be subject to regulatory compliance requirements. For example, if the software is intended for use in food production, it may be subject to regulations governing food safety and quality.

5.3 Security Requirements

- Data Privacy: The software should ensure that the images of the fruit that are captured and used for the maturity detection process are kept confidential and not accessible to unauthorized users.

- Authentication: The software should include a user authentication mechanism to ensure that only authorized users can access the system and the data within it

- Data Integrity: The software should ensure that the data is not tampered with or modified in any way that could affect the accuracy of the maturity detection process.

- Encryption: The software should use encryption to secure the data at rest and in transit to prevent unauthorized access or interception.

- Secure Communication: The software should use secure communication protocols to ensure that data is transmitted securely and not accessible to unauthorized users.

- Auditability: The software should have an audit trail to track all changes made to the system and the data within it, which can be used to detect any potential security breaches.

- Vulnerability Testing: The software should be tested regularly to identify any vulnerabilities or weaknesses that could be exploited by malicious actors.

5.4 Software Quality Attributes

The following software quality attributes should be considered:

- Accuracy: This refers to the ability of the software to accurately detect the maturity level of fruits. The accuracy of the software should be high to ensure that the results produced are reliable.

- Reliability: This refers to the ability of the software to consistently produce accurate results over time. The software should be reliable and stable, with minimal errors or glitches.
- Efficiency: This refers to the speed and performance of the software. The software should be able to process images quickly and efficiently, without causing any delays or lag.
- Usability: This refers to the ease of use of the software. The software should be user-friendly, with a simple and intuitive interface that allows users to easily input and output data.
- Maintainability: This refers to the ease with which the software can be maintained and updated. The software should be designed in a modular and flexible manner, with clearly defined interfaces and separation of concerns, making it easy to modify or add new functionality.
- Scalability: This refers to the ability of the software to handle an increasing number of inputs and users over time. The software should be designed to be scalable, with the ability to handle large volumes of data and users without any performance issues.
- Security: This refers to the ability of the software to protect sensitive data and prevent unauthorized access. The software should be designed with robust security features that ensure the privacy and confidentiality of user data.

To document these software quality attributes for Fruit Maturity Detection software using Convolutional Neural Networks, the following steps can be taken:

- Define each quality attribute and its relevance to the software.
- Determine the metrics that will be used to measure each quality attribute.
- Identify the tools and techniques that will be used to monitor and evaluate each quality attribute.
- Develop a plan for testing and validating each quality attribute.
- Document the results of testing and evaluation, along with any recommendations for improvements.
- Continuously monitor and evaluate the software to ensure that it continues to meet the desired quality attributes.

5.5 Business Rules

Fruit maturity detection is a process of determining the ripeness of a fruit. It is an important factor in the food industry, as it affects the quality of the fruit and the products derived from it. Convolutional Neural Networks (CNNs) are a popular machine learning technique used in image recognition tasks. They can be used in fruit maturity detection to recognize the maturity level of a fruit based on its visual appearance. This document outlines the business rules of a fruit maturity detection software using CNN.

Business Rules:

- The software should be able to detect the maturity level of a fruit based on its visual appearance.

- The software should be trained on a dataset of images of fruits at different stages of maturity.
- The software should be able to classify a fruit into one of the following maturity levels: unripe, ripe, or overripe.
- The software should have an accuracy of at least 90% in detecting the maturity level of a fruit.
- The software should be able to detect the maturity level of a variety of fruits, including but not limited to, apples, bananas, oranges, and grapes.
- The software should be able to detect the maturity level of a fruit regardless of its shape, size, or color.
- The software should be able to provide a visual representation of the detected maturity level of a fruit.
- The software should be able to store the data of the detected fruit maturity levels for future analysis and reporting.
- The software should have a user-friendly interface for easy operation.

6. Other Requirements

- High-quality image data: The CNN algorithm requires high-quality image data to learn and make accurate predictions. This means that the images used in the training dataset should be clear, high resolution, and free from any noise or distortion.
- Annotated dataset: An annotated dataset is required for training the CNN. This means that each image in the dataset should have a label indicating the level of maturity of the fruit. This can be done manually by experts or using automated tools such as image processing algorithms.
- Pre-processing of images: Pre-processing of images is essential to remove any background noise and to enhance the image contrast. This step can include various operations such as filtering, thresholding, and normalization.
- CNN architecture: The CNN architecture should be chosen based on the type of fruit and the complexity of the task. A deeper network with more layers may be required for more complex fruits such as pineapples, while a shallower network may be sufficient for simpler fruits such as bananas.

- Training parameters: The training parameters such as the learning rate, number of epochs, and batch size should be optimized to achieve the best accuracy. This requires extensive experimentation and tuning of the parameters.

- Integration with hardware: The software should be integrated with suitable hardware such as cameras and sensors for capturing the images of the fruits. The hardware should be capable of capturing high-quality images in a consistent and reliable manner.

- Real-time processing: The software should be designed to process the images in real-time, as fruits are often inspected on a conveyor belt or in a production line. This requires efficient algorithms and hardware that can process the images quickly and accurately.

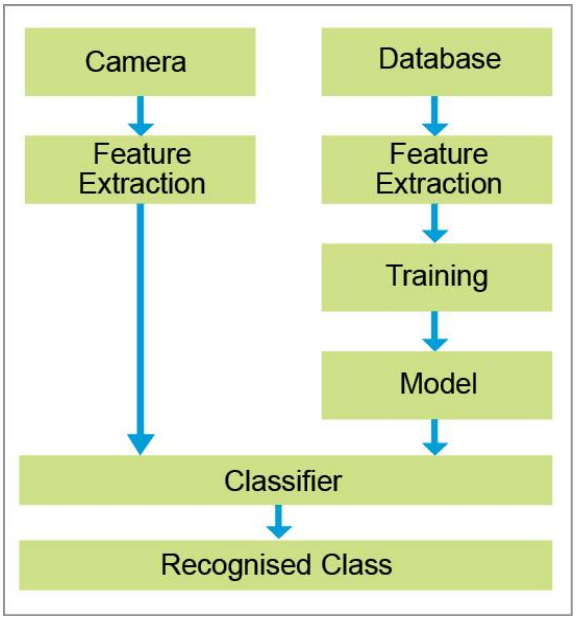
- User interface: The software should have a user-friendly interface that allows users to view the results, adjust the parameters, and perform other tasks easily. The interface should be intuitive and easy to use, even for non-experts.

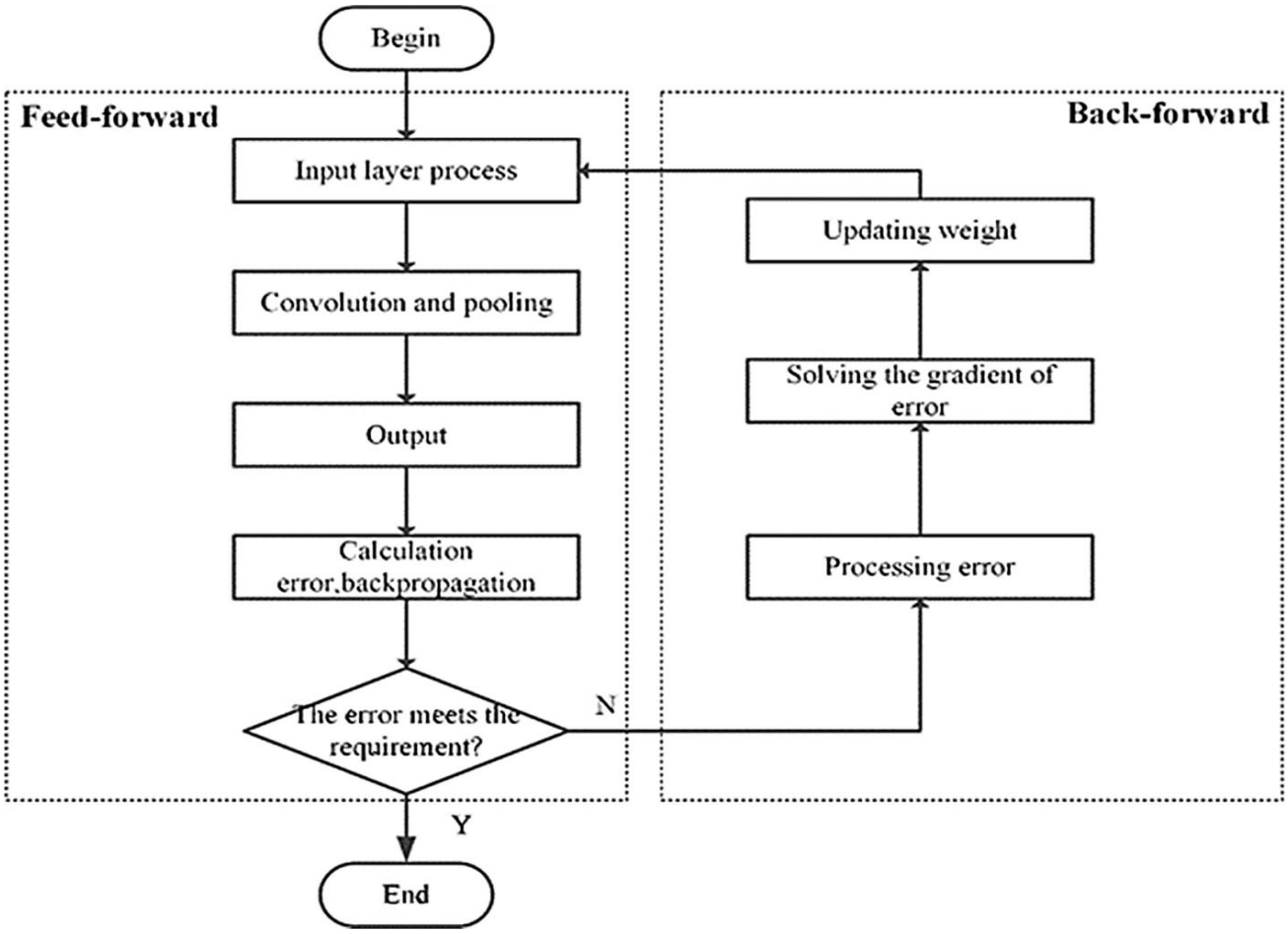
Appendix A: Glossary

- Fruit Maturity: The stage of fruit development where it has reached its optimal level of ripeness and readiness for consumption.
- Convolution Neural Network (CNN): A deep learning algorithm that is particularly well-suited for image recognition and classification.
- Pre-processing: The steps taken to prepare the input data for the CNN, including image resizing, normalization, and augmentation.
- Model Architecture: The structure of the CNN, which includes the number and arrangement of layers and neurons, activation functions, and loss functions.
- Training Data: A set of labelled images used to train the CNN to recognize patterns and classify images.
- Validation Data: A set of labelled images used to validate the performance of the CNN during training and optimize its hyperparameters.
- Test Data: A set of unlabelled images used to test the accuracy of the trained CNN in identifying fruit maturity.
- Hyperparameters: The parameters of the CNN that are set prior to training, including learning rate, batch size, and number of epochs.
- Loss Function: A mathematical function that measures the difference between the predicted and actual labels of the training data, used to optimize the CNN during training.
- Optimization Algorithm: An algorithm used to minimize the loss function during training, such as Stochastic Gradient Descent or Adam.
- Backpropagation: The process of computing the gradients of the loss function with respect to the CNN's parameters, used to update the weights of the CNN during training.
- Accuracy: A measure of the CNN's performance in correctly classifying fruit maturity, calculated as the percentage of correctly classified images in the test data.

Appendix B: Analysis Models

The fruit maturity detection software using Convolutional Neural Network is a software system that uses AI and machine learning techniques to identify the maturity level of fruits. It is based on the Convolutional Neural Network (CNN) model, which is a deep learning algorithm that is commonly used for image recognition tasks.
The system takes images of fruits as input and uses the CNN algorithm to analyze the image features and identify the maturity level of the fruit. The software has been trained on a dataset of images that contain different levels of fruit maturity, and it uses this data to learn and improve its accuracy.
The analysis model of this software consists of several layers of neural networks that perform various tasks, such as image feature extraction, image classification, and prediction. The input layer takes the raw image data as input, and the output layer produces the prediction of the fruit maturity level.
The CNN model used in this software is designed to handle large datasets and complex image features. It uses convolutional layers to extract features from the input image, and pooling layers to reduce the dimensionality of the feature map. These features are then fed into fully connected layers, which perform the classification and prediction tasks.
Overall, the Fruit Maturity Detection software using Convolutional Neural Network is a powerful tool for identifying the maturity level of fruits. It can be used in various applications, such as in the agriculture industry for monitoring the growth and quality of fruits, or in the food industry for ensuring the freshness and quality of fruits.





Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>