Abstract

Ensuring fruit quality is essential for consumer safety and waste reduction. Our system utilizes Convolutional Neural Networks(CNN) and transfer learning to classify fruits as good or bad based on image analysis. By leveraging pre-trained deep learning models, the system efficiently extracts key features such as texture, color, and defects to determine fruit suitability for consumption. This automated approach enhances accuracy, speed, and reliability in fruit quality assessment. With a user-friendly interface and real-time processing capabilities, the system provides a practical solution for farmers, vendors, and consumers, promoting better decision-making in food supply chains.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Overview

Our system provides an efficient and automated solution for assessing fruit quality, helping farmers, vendors, and consumers make informed decisions. Using Convolutional Neural Networks (CNN) and transfer learning, it classifies fruits as good or bad based on visual features like texture, color, and surface defects.By leveraging deep learning models, the system ensures high accuracy and real-time analysis, reducing the reliance on manual inspection. Designed for ease of use and scalability, it can be integrated into various agricultural and retail settings, promoting better quality control, minimizing food waste, and ensuring consumer safety.

1.2 General Background

Our fruit quality detection system is designed to provide fast, accurate, and automated assessment of fruit freshness and suitability for consumption. Traditional methods of fruit inspection rely on manual observation, which can be time-consuming and inconsistent. To address this challenge, our system utilizes Convolutional Neural Networks (CNN) and transfer learning to classify fruits as good or bad based on key visual features such as color, texture, and surface defects.By analyzing fruit images with deep learning models, the system enhances efficiency and reliability in quality assessment. It can be integrated into various agricultural and retail settings, helping farmers, vendors, and consumers make better purchasing and selling decisions. This solution supports waste reduction, food safety, and improved quality control, ensuring that only fresh and high-quality produce reaches the market.

1.3 Problem statement

Ensuring fruit quality is a crucial challenge in the agricultural and retail industries, as traditional inspection methods rely on manual observation, which can be timeconsuming, inconsistent, and prone to errors. Farmers, vendors, and consumers often struggle to determine whether a fruit is fresh or spoiled, leading to issues like food waste, financial losses, and compromised food safety. This project aims to develop an AI-powered Fruit Quality Detection System that utilizes Convolutional Neural Networks (CNN) and transfer learning to accurately classify fruits as good (fresh) or bad (spoiled/rotten). By leveraging machine learning and image processing, the system provides an automated, fast, and reliable solution for fruit quality assessment. With a user-friendly interface that allows users to upload images for real-time analysis, this system helps improve quality control, reduce food waste, and ensure safer consumption.

1.4 Scope of the System

- Image Upload Analysis: Users can upload images of fruits through the system's web interface. The system then processes these images using Convolutional Neural Networks (CNN) and transfer learning to extract key visual features such as color, texture, and surface defects for quality assessment.

- Fruit Classification: The system analyzes the uploaded fruit image and classifies it as either good (fresh) or bad (spoiled/rotten) based on deep learning predictions.This automated classification provides a fast, accurate, and reliable alternative to traditional manual inspection methods.

- Administrative Management:Administrators oversee the system by managing fruit datasets, updating classification models, and optimizing the detection algorithms.They ensure data accuracy, model efficiency, and system scalability, making necessary adjustments to enhance overall performance and reliability.

1.5 Objective

The objective of this system is to efficiently assess fruit quality by leveraging machine learning and computer vision to classify fruits as good (fresh) or bad (spoiled/rotten).By utilizing Convolutional Neural Networks (CNN) and transfer learning, the system ensures accurate and automated fruit inspection, reducing reliance on manual quality checks and minimizing human error.This project aims to enhance food safety, reduce waste, and improve quality control by providing a fast, reliable, and user-friendly solution for farmers, vendors, and consumers.By maintaining a structured database of fruit images and classifications, the system continuously improves its detection accuracy, making it adaptable to various fruit types and conditions. Ultimately, this solution promotes better decision-making in agriculture and retail, ensuring that only fresh, high-quality produce reaches the market.

Chapter 2

Literature Review

2.1 Enhanced CNN for Fruit Disease Detection and Grading Classification Using SSDAE-SVM for Postharvest Fruit

2.1.1 Abstract

Fruit disease detection and grading are crucial for ensuring high-quality produce and minimizing postharvest losses. This study presents an enhanced Convolutional Neural Network (CNN) with Spatial Pyramid Pooling (SPP) for accurate disease identification.To improve image quality, an Optimum FIR Wiener Filter is applied for noise reduction. Feature extraction focuses on fruit attributes such as color,shape,and texture.For fruit grading, a Stacked Sparse Denoising Autoencoder (SSDAE) combined with Support Vector Machine (SVM) efficiently classifies fruits into healthy and unhealthy categories. The proposed system achieves 97.25

2.1.2 Methodology

The proposed system utilizes an advanced Convolutional Neural Network (CNN) with Spatial Pyramid Pooling (SPP) to enhance disease detection by extracting key fruit features such as color, shape, and texture.To ensure high-quality input images, an Optimum FIR Wiener Filter is applied during preprocessing, effectively reducing noise and improving image clarity.This step enhances the accuracy of disease identification by ensuring that only relevant and refined image details are processed. For fruit grading, a Stacked Sparse Denoising Autoencoder (SSDAE) is used to extract high-dimensional features, which are then classified using a Support Vector Machine (SVM).This hybrid approach ensures precise categorization of fruits into healthy and unhealthy categories, aiding in better postharvest quality control.By integrating deep learning and machine learning techniques, the system achieves high accuracy while maintaining efficiency in image processing and classification.

2.1.3 Result Analysis

The proposed CNN-based fruit disease detection and grading system provides an efficient and accurate solution for postharvest fruit inspection.By integrating Spatial Pyramid Pooling (SPP) with CNN, the model enhances feature extraction for precise disease identification, while the Optimum FIR Wiener Filter ensures high-quality image preprocessing. The SSDAE-SVM approach further improves fruit grading by effectively classifying healthy and unhealthy fruits. With an accuracy of 97.25

2.2 Deep Learning Model For Fruit Quality Detection And Evaluation

2.2.1 Abstract

Ensuring fruit quality is essential for maintaining food safety and reducing waste in the agricultural industry.This study presents a deep learning-based fruit quality detection system that leverages Convolutional Neural Networks (CNNs), specifically ResNet-50, to assess fruit freshness and classify fruits based on quality.The system evaluates multiple pre-trained models and identifies ResNet-50 as the most suitable due to its high accuracy and robust feature extraction capabilities.The dataset consists of 12,069 images, categorized into Good, Bad, and Mixed quality classes, with preprocessing techniques applied for noise reduction. By automating the fruit quality evaluation process, this model enhances efficiency in food supply chains.However, its performance is dependent on image quality and requires high computational resources for real-time deployment.Future work aims to improve computational efficiency and explore lightweight models for broader applicability.

2.2.2 Methodology

The proposed system employs ResNet-50, a deep Convolutional Neural Network (CNN), for feature extraction and classification of fruit quality.The dataset comprises 12,069 images, divided into training and testing sets, ensuring a robust evaluation of the model's performance.Images are preprocessed and categorized into Good, Bad, and Mixed quality classes, with filtering techniques applied to enhance clarity.The ResNet-50 model is trained using these processed images, leveraging its deep architecture to extract intricate patterns and features essential for accurate classification. The training process is optimized to minimize errors and improve detection accuracy.This methodology enables efficient and automated fruit quality assessment, making it suitable for large-scale applications in the food industry.

2.2.3 Result Analysis

This study demonstrates the effectiveness of ResNet-50 in fruit quality detection, offering high accuracy and automated classification to improve postharvest quality control.By leveraging deep learning, the system efficiently distinguishes between different fruit quality categories, reducing human intervention and enhancing reliability.However, the model's reliance on high-quality images and computational resources poses challenges for real-time and large-scale applications.Future research can focus on optimizing computational efficiency, exploring lightweight deep learning models, and integrating real-time edge computing solutions to enhance scalability and accessibility in agricultural and food industries.

2.3 Fruit quality detection and classification using Computer Vision Techniques

2.3.1 Abstract

Ensuring fruit quality and safety is a critical concern in the food industry, particularly with increasing pesticide contamination. This study proposes a computer vision-based system that integrates deep learning, IoT sensors, and cloud computing for real-time fruit quality detection and classification. The system employs Deep Learning models with Keras and TensorFlow to classify fruits based on their quality and detect contamination. Additionally, IoT sensors (humidity, temperature, and gas sensors) measure pesticide levels, enabling early detection of harmful substances. Transfer Learning techniques enhance image recognition and classification accuracy. By leveraging cloud computing, the system provides real-time monitoring, ensuring food safety and efficient quality control. However, challenges such as internet dependency and sensor accuracy fluctuations must be addressed for large-scale implementation.

2.3.2 Methodology

The proposed system integrates deep learning and IoT-based monitoring for fruit quality assessment. Keras and TensorFlow are used to develop a deep learning model, enabling accurate fruit classification through feature extraction and pattern recognition. IoT sensors, including humidity, temperature, and gas sensors, detect pesticide contamination and environmental factors affecting fruit quality. The collected data is processed using cloud computing, allowing real-time monitoring and decision-making. To enhance accuracy and efficiency, Transfer Learning techniques are implemented, leveraging pre-trained models for improved image recognition. This methodology ensures a cost-effective, scalable, and automated approach to fruit quality detection while addressing contamination risks.

2.3.3 Result Analysis

This study presents an IoT and deep learning-powered fruit quality detection system, offering real-time monitoring and accurate classification of fruits while detecting pesticide contamination. The integration of cloud computing and IoT sensors ensures a practical and scalable solution for food safety. Despite its advantages, challenges such as internet connectivity dependency and sensor accuracy variations must be addressed to enhance reliability. Future research can focus on improving sensor calibration, optimizing model efficiency, and developing offline processing capabilities to make the system more adaptable and effective in diverse agricultural environments.

Chapter 3

Requirement Specification

3.1 Functional Requirement

- Image upload: Users should be able to upload fruit images in formats like JPEG, PNG, etc. for analysis.

- Image upload: Users should be able to upload fruit images in formats like JPEG, PNG, etc. for analysis.

- Fruit Quality Classification: The system should analyze the uploaded image and classify the fruit as good or bad using CNN and transfer learning.

- Edibility Assessment: Based on the classification, the system should determine whether the fruit is suitable for consumption or not.

3.2 Software Requirement

- Frontend: React.js provides a dynamic and interactive user-friendly interface, allowing users to upload fruit images for quality detection. Node.js serves as the backend environment to manage API requests and handle communication with the server efficiently.

- Backend: Python is used for processing the uploaded images and performing fruit quality detection using machine learning (CNN). It offers powerful libraries such as TensorFlow, Keras, and OpenCV, making it an ideal choice for handling image data..

3.3 Hardware Requirement

Laptop

3.4 Hardware Specifications

Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz

RAM: 8.00 GB

Storage: 1TB+ SSD

Internet: Medium-speed connection

Chapter 4

Proposed system and Design

This chapter is mainly discuss about the proposed system and design.Also the architecture and differential technical diagrams are discussed in this chapter.

4.1 Proposed system

The fruit quality detection system is designed to assess the freshness of fruits using machine learning and image processing. The system enables users to upload fruit images, which are analyzed using Convolutional Neural Networks (CNN) and transfer learning to classify them as good (fresh) or bad (spoiled/rotten). By extracting key visual features such as color, texture, and surface defects, the model ensures accurate and reliable quality assessment, helping to reduce food waste and enhance consumer safety.

The system is trained on a dataset containing diverse fruit images labeled with their respective quality conditions. This dataset helps the model learn from varied lighting conditions, different fruit types, and multiple stages of ripeness or decay, improving classification accuracy.

To provide a seamless user experience, the system features a web-based interface developed using React.js and Node.js, allowing users to upload images and receive real-time classification results. The backend, powered by Python with TensorFlow, Keras, and OpenCV, processes the images efficiently to generate accurate predictions.

Beyond classification, the system can be expanded with additional features, such as early spoilage detection, storage recommendations, and integration with supply chain management for large-scale agricultural applications.This system serves as a valuable tool for consumers, retailers, and agricultural industries in ensuring fruit quality and minimizing waste.

4.2 Feasibility Study

The proposed Fruit Quality Detection System is a viable solution that leverages machine learning and image processing to classify fruits as good (fresh) or bad (spoiled/rotten). By using AI-based analysis, the system enhances food safety, reduces waste, and streamlines quality assessment for consumers, retailers, and agricultural industries.

4.2.1 Technical Feasibility

The front end is developed using React.js for a dynamic and interactive user-friendly interface, allowing users to upload fruit images for quality detection. The back end is built using Node.js to efficiently handle API requests and server communication. Python is used for image processing and machine learning (CNN), leveraging libraries like TensorFlow, Keras, and OpenCV. Since these technologies are widely used and well-documented, the project is technically feasible.

4.2.2 Operational Feasibility

Compared to existing datasets and models, this system provides a more convenient and automated approach* to fruit quality assessment. Advanced machine learning techniques and image processing ensure accurate fruit classification and defect detection. This makes the system operationally feasible for both agriculture and food industries. operational feasibility.

4.2.3 Economic Feasibility

The project utilizes open-source technologies such as React.js, Node.js, Python, TensorFlow, and OpenCV, eliminating licensing costs. This makes development cost-effective while maintaining high efficiency and accuracy. Thus, the project is economically feasible.

4.3 Design

4.3.1 Architecture Diagram

The system enables users to upload fruit images, processes data for quality assessment using Convolutional Neural Networks (CNN) and transfer learning, and classifies fruits as good or bad based on visual features like color, texture, and surface defects. This automated approach ensures accurate and efficient fruit quality detection, helping to reduce food waste and ensure consumer safety.

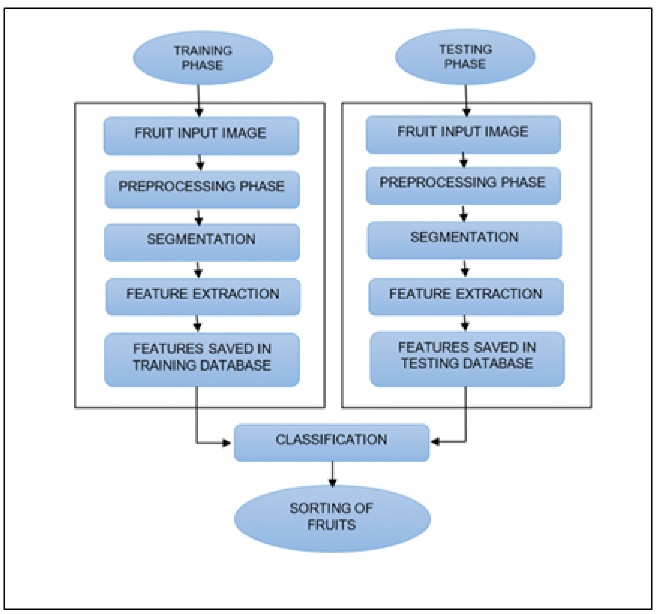

Figure 4.1: Architecture Diagram

4.3.2 Use Case Diagram

The Fruit Quality Detection System lets users upload fruit images for analysis. Using CNN-based models, it detects the fruit, analyzes its features, and classifies it as fresh or spoiled. The system then displays the result, ensuring accurate and automated fruit quality assessment to enhance food safety. This helps reduce waste and supports better decision-making in the food industry.

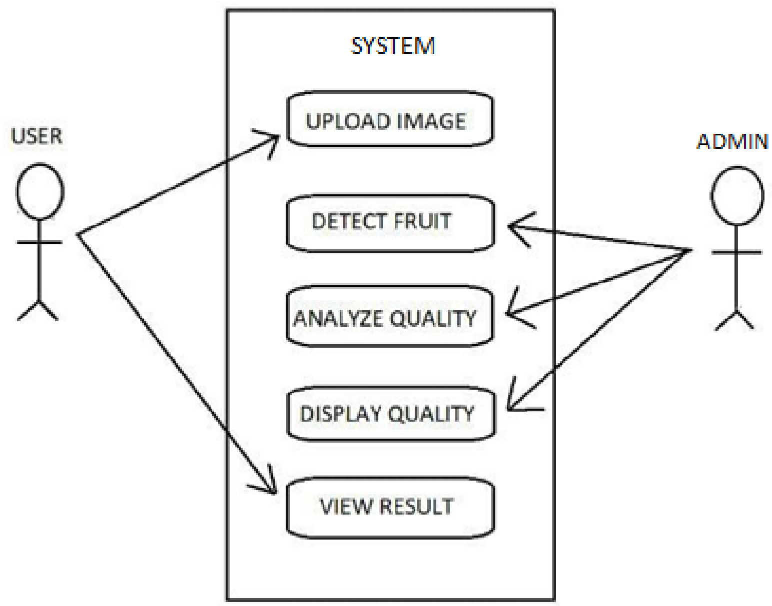

Figure 4.2: Use Case Diagram

4.3.3 Data Flow Diagram

The system takes fruit images as input, along with pre-trained data for classification. A Convolutional Neural Network (CNN) processes these inputs to classify the fruit based on quality parameters. The system receives user input, which undergoes preprocessing to enhance image quality. The preprocessed image is then fed into the CNN model, which classifies the fruit as good or bad based on learned features. The system then displays the classification results to the user, helping in making informed decisions about fruit freshness.

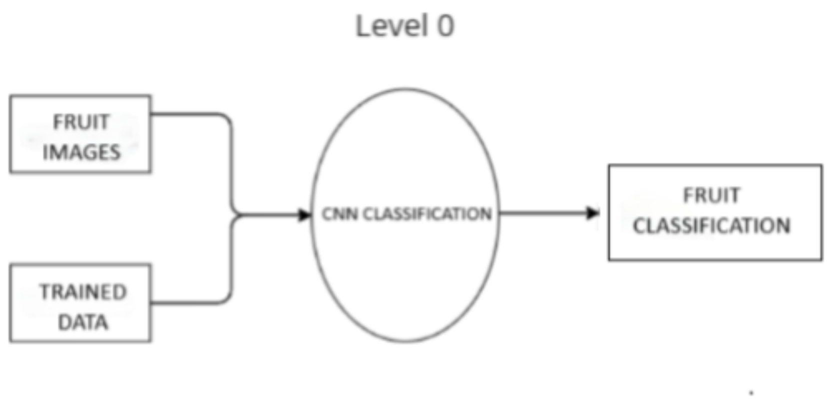

Figure 4.3: Level 0

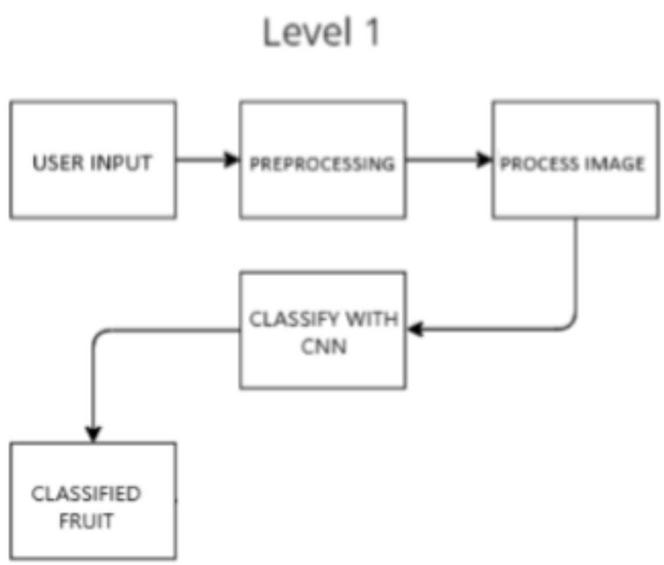

Figure 4.4: Level 1

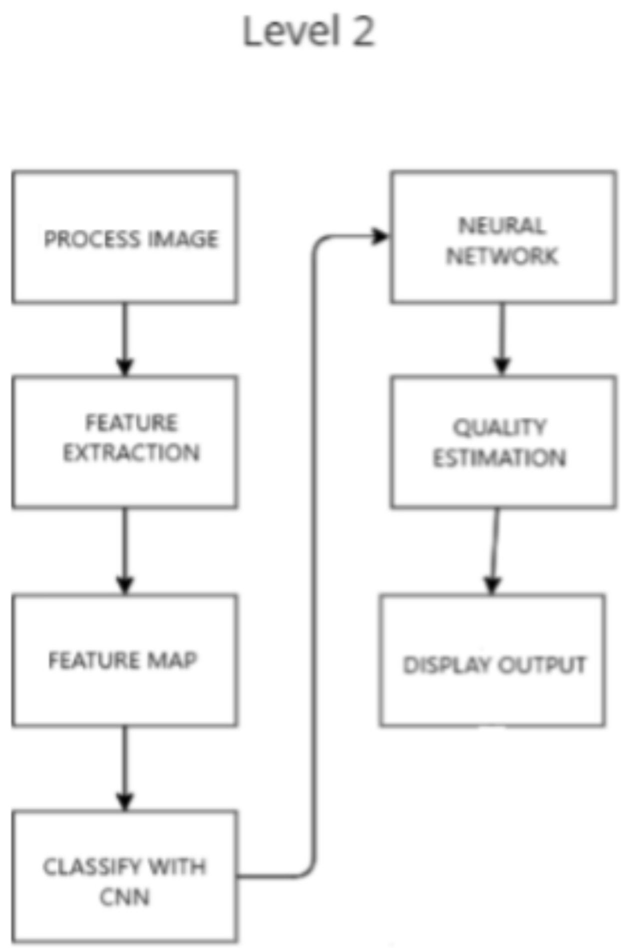

Figure 4.5: Level 2

4.4 Gantt Chart

This Gantt chart provides a concise timeline for a project, outlining tasks over three months (December to February):

1. Task 1: Problem Statement (December)

2. Task 2: Data Collection (December–January)
3. Task 3: Data Training (January)
4. Task 4: Model Testing (February)

It efficiently visualizes tasks for clear project planning.

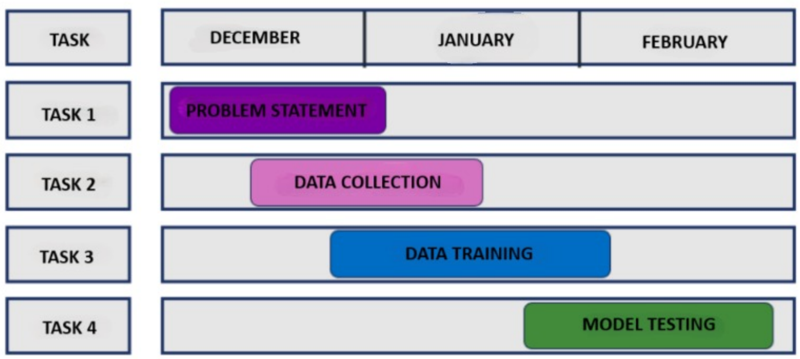

Figure 4.6: Modules Split-up and Gantt Chart

Chapter 5

Implementation

5.1 Implementation Details

5.1.1 Load the trained model and define class model

```
model = tf.keras.models.load_model("fruit_classifier_model.h5")

class_labels = ["Good Orange", "Bad Orange", "Good Apple", "Bad Apple", "Good Pomegranate", "Bad Pomegranate"]
```

Figure 5.1: Trained Model

5.1.2 Preprocess and predict image

```
def predict_fruit(image_path):
    IMG_SIZE = 100
    img = cv2.imread(image_path)
    if img is None:
        print("Error: Unable to load image.")
        return

    img = cv2.resize(img, (IMG_SIZE, IMG_SIZE))
    img = img / 255.0
    img = np.expand_dims(img, axis=0)

    prediction = model.predict(img)
    predicted_class = np.argmax(prediction)
    confidence = np.max(prediction) * 100
```

Figure 5.2: Prediction Values

5.1.3 Determine if the fruit is good or bad

```
if "Bad" in class_labels[predicted_class]:
    fruit_quality = "Bad"
else:
    fruit_quality = "Good"
```

Figure 5.3: Quality Detection

5.1.4 Display Image and Prediction

```
plt.imshow(cv2.cvtColor(cv2.imread(image_path), cv2.COLOR_BGR2RGB))
plt.axis("off")
plt.title(f"Prediction: {fruit_quality} ({confidence:.2f}%)")
plt.show()
```

Figure 5.4: Display Image

5.2 Modules Used

5.2.1 Capturing and Processing Images:- Library: OpenCV (cv2)

- OpenCVis used for reading, resizing, and processing images.
- Functions used:
  - cv2.imread(imagepath): Reads an image from the given path.
  - cv2.resize(img, (IMGSIZE, IMGSIZE)): Resizes the image to match the model's input size.
  - cv2.COLORBGR2RGB: Converts an image from BGR to RGB format for correct color representation in matplotlib.

5.2.2 NumericalComputationandDataHandling:-Library: NumPy (numpy)

- NumPyis used for numerical operations and handling image data as arrays.

- Functions used:
  - np.expanddims(img, axis=0): Adds an extra dimension to match the model's expected input shape.
  - np.argmax(prediction): Finds the index of the highest predicted probability.
  - np.max(prediction): Gets the maximum probability value to determine confidence.

5.2.3 DeepLearningModelLoadingandPredictionLibrary:-TensorFlow/Keras (tensorflow)

- TensorFlow is used for loading and running a trained Convolutional Neural Network (CNN) model.
- Functions used:
  - tf.keras.models.loadmodel("fruitclassifiermodel.h5"): Loads the pre-trained deep learning model.
  - model.predict(img): Makes a prediction based on the input image

5.2.4 File Handling and Path ManagementLibrary:- OS (os)

- OS is used for handling file paths and directories (not explicitly used in the script but useful for managing datasets dynamically)

5.2.5 Visualization and Display of Results :-Library: Matplotlib (matplotlib.pyplot)

- Matplotlib is used for displaying images and classification results.
- Functions used:
  - plt.imshow(cv2.cvtColor(cv2.imread(imagepath), cv2.COLORBGR2RGB)): Displays the image with correct color format.

- plt.axis(”off”): Hides the axis for a cleaner display.
- plt.show(): Displays the image with the prediction overlay.

Chapter 6

Result and Discussion

6.1 Results

6.1.1 Good Banana

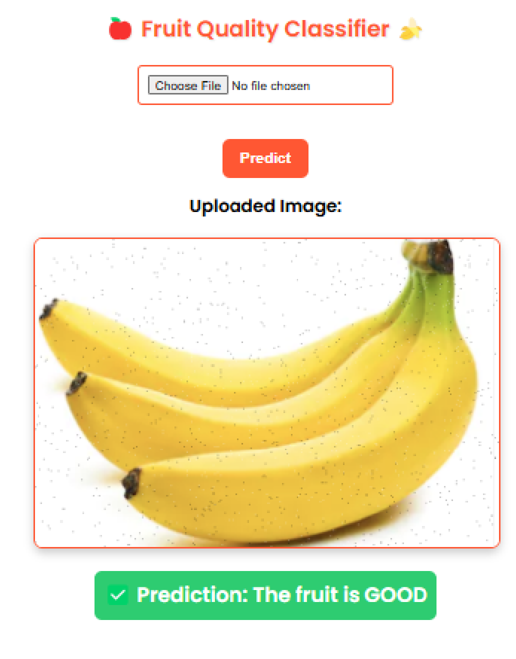

Figure 6.1: Sample Output

6.1.2 Bad Strawberry

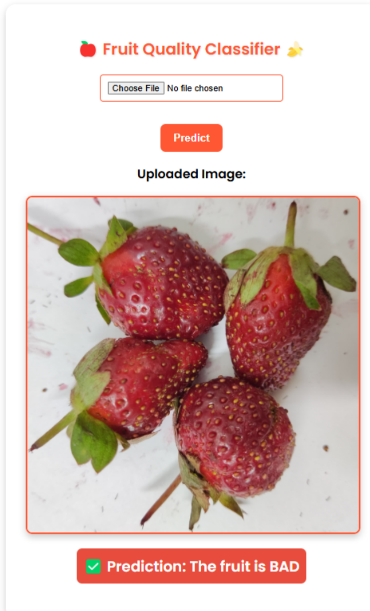

Figure 6.2: Sample Output

6.1.3 Good Orange

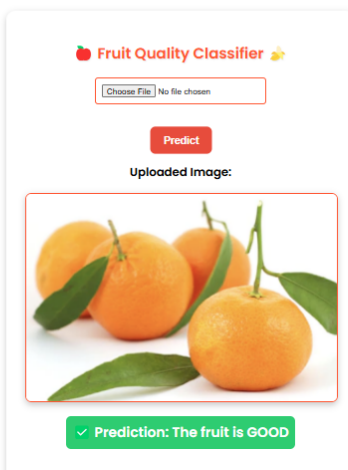

Figure 6.3: Sample Output

6.1.4 Bad Mango

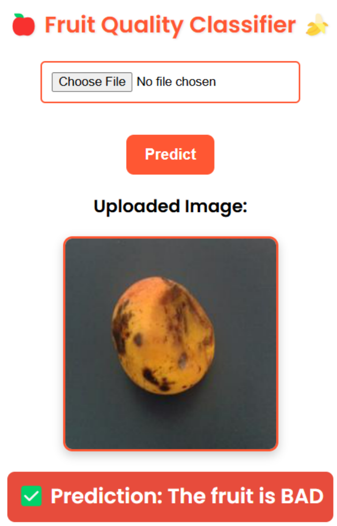

Figure 6.4: Sample Output

Chapter 7

Conclusion

The fruit quality detection system effectively leverages machine learning and image processing to classify fruits as good (fresh) or bad (spoiled/rotten).By utilizing Convolutional Neural Networks (CNN) and transfer learning, the system ensures accurate classification based on key visual features such as color, texture, and surface defects.With a user-friendly web-based interface developed using HTML and a Python-powered backend utilizing TensorFlow, Keras, and OpenCV, the system provides real-time results, enhancing efficiency in quality assessment.

This automated approach reduces reliance on manual inspection, minimizes human error, and improves food safety by identifying spoiled fruits early. Its applicability extends to households, grocery stores, and large-scale agricultural industries, reducing food waste and streamlining postharvest processing. Although factors like image quality and lighting variations may impact accuracy, these challenges can be mitigated through dataset expansion and model optimization. Further improvements can focus on enhancing the system's processing speed, optimizing it for lower-end devices, and incorporating multi-class classification to assess various quality levels. Additionally, integrating features such as predictive analysis for shelf-life estimation and customized recommendations for storage conditions can further improve its usability. By continuing to refine and expand its capabilities, the system can play a vital role in ensuring high-quality fruit distribution and consumption

References

1. **HimanshuB.Patel and NitinJ.Patil** *"Enhanced CNN for Fruit Disease Detection and Grading Classification." Using SSDAE-SVM for Postharvest Fruits* IEEE SENSORS JOURNAL,VOL.24,NO.5,1 MARCH 2024

2. **Swapnil Jadhav, Janhavi Naik, Prawit Chumchu** *"DEEP LEARNING MODEL FOR FRUIT QUALITY DETECTION AND EVALUATION" Volume: 9— Issue: 5— May 2023—— Journal DOI: 10.36713/epra2013 ——SJIF Impact Factor 2023: 8.224 —— ISI Value: 1.188*

3.**T. Thomas Leonid, Hemamalini, Nanthine T, Krithika** *"Fruit quality detection and classification using Computer Vision Techniques" 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM) — 979-8-3503-4779 1/23/31.002023IEEE—DOI : 10.1109/ICONSTEM56934.2023.10142514*