FRUIT QUALITY DETECTION USING DEEP LEARNING FOR ROTTEN AND FRESH FRUITS CLASSIFICATION

A PROJECT REPORT

Submitted by

PRABHAKARAN M

(731223622009)

in partial fulfillment for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

J.K.K MUNIRAJAH COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

T.N. PALAYAM, GOBI-638 506.

ANNA UNIVERSITY, CHENNAI 600 025

MAY 2025

BONAFIDE CERTIFICATE

ANNA UNIVERSITY, CHENNAI 600 025

Certified that this Report titled "**FRUIT QUALITY DETECTION USING DEEP LEARNING FOR ROTTEN AND FRESH FRUITS CLASSIFICATION**" is the Bonafide work of " **PRABHAKARAN M** " **(731223622009)**" who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Dr.N. SATHYABALAJI., M.E., Ph.D.	**Dr.N. SATHYABALAJI., M.E., Ph.D.**
Associate Professor,	Associate Professor,
Dept. Of Computer Applications**,**	Dept. Of Computer Applications**,**
J.K.K Munirajah College of Technology,	J.K.K Munirajah College of Technology,
T.N. Palayam.	T.N. Palayam.

Submitted for the Project Viva-voce Examination held on……………………….

_______________________	____________________
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

I express my sincere thanks and grateful acknowledgment to our chairman **Dr. J.K.K. MUNIRAJAH M. TECH (Bolton)., D.Litt.,** for providing all facilities during the course of study in this college.

I would like to express our thanks to our secretary madam **Mrs. KASTHURIPRIYA KRUPAKARMURALI M.B.A.,** who has provided all the available facilities and support that has held me in completing the project.

I have immense pleasure in expressing my extreme gratitude thanks to our beloved principal **Dr. K. SRIDHARAN M.E., MBA., Ph.D., M.I.S.T.E.,** for his encouragement and support.

I wish to express my heartful thanks to our respectful Head of the Department **Dr. N. SATHYABALAJI M.E., M.I.S.T.E., Ph.D.,** for his inspiring help, guidance, effort and energy in the right direction for completing this project.

I am privileged to thank and express my extreme gratitude to our project coordinator **Dr. N. SATHYABALAJI M.E., M.I.S.T.E., Ph.D.,** for his enlightening thoughts and remarkable guidance that helped me in the successful completion of the project.

I also thank my guide **Dr. N. SATHYABALAJI M.E., M.I.S.T.E., Ph.D.,** who has been a driving force to unveil the immense talents in me.

I sincerely thank my lovable parents for their motivation and great support to complete this project successfully.

I also thank all teaching and non-teaching staffs of the Department of Computer Science and Engineering and all my friends for their help and support to complete this project successfully.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

	ABBREVIATIONS
DCNN	DEEP CONVOLUTIONAL NEURAL NETWORK
TTA	TEST - TIME AUGMENTATION
JPG	JOINT PHOTOGRAPHIC EXPERTS GROUPS
IOT	INTERNET OF THINGS
SMQTT	SECURE MESSAGE QUEUE TELEMETRY TRANSPORT
DL	DEEPL LEARNING
SVM	SUPPORT VECTOR MACHINE
PNG	PORTABLE NETWORK GRAPHICS
GNB	GAUSSIAN NAÏVE BAYES
CNN	CONVOLUTIONAL NEURAL NETWORK
NIC	NETWORK INTERFACE CARD
HTML	HYPER TEXT MARKUP LANGUAGE
SVG	SCALABLE VECTOR GRAPHICS
GSM	GLOBAL SYSTEM FOR MOBILE
JSON	JAVASCRIPT OBJECT NOTATION
PDF	PORTABLE DOCUMENT FORMAT
CLI	COMMAND LINE INTERFACE
GUI	GRAPHICAL USER INTERFACE
TTA	TRAINING AND TECHNICAL ASSISTANCE
RGA	RETURN GOODS AUTHORIZATION

ABSTRACT

This project introduces a deep learning-based web application designed to classify fruit images into six categories: fresh apple, fresh banana, fresh orange, rotten apple, rotten banana, and rotten orange. The classification system is powered by the EfficientNetB1 model, a lightweight yet powerful convolutional neural network known for its optimized performance and accuracy. The model has been trained to detect subtle visual cues in fruit images that distinguish freshness from decay, making it suitable for quality control tasks in agricultural and retail environments.

To improve prediction accuracy and generalization, especially when handling rotated or slightly varied input images, the system incorporates Test-Time Augmentation (TTA). This technique applies random image transformations, such as rotation, during the inference stage and averages multiple prediction results to produce a more reliable output. This augmentation helps mitigate challenges posed by diverse lighting, angles, and background variations that are common in image data. The model outputs the top predicted class along with a confidence score and also lists the top three predictions to provide more insight into its decision-making process.The application interface is built using Stream lit, offering a clean and intuitive user experience. Users can upload images of individual fruits in formats such as JPG or PNG and receive instant classification results. The system also includes progress bars and helpful visual cues to enhance interpretability. With its combination of an advanced neural architecture and TTA-based robustness, this classifier demonstrates the practical application of deep learning in the field of food technology and agricultural automation.

சுருக்கம்

PNG போன்ற வடிவங்களில் தனிப்பட்ட பழங்களின் படங்களை இந்தத் திட்டம் பழப் படங்களை ஆறு வகைகளாக வகைப்படுத்த வடிவமைக்கப்பட்ட ஒரு ஆழமான கற்றல் அடிப்படையிலான வலை பயன்பாட்டை அறிமுகப்படுத்துகிறது: புதிய ஆப்பிள், புதிய வாழைப்பழம், புதிய ஆரஞ்சு, அழுகிய ஆப்பிள், அழுகிய வாழைப்பழம் மற்றும் அழுகிய ஆரஞ்சு. வகைப்பாடு அமைப்பு திறமையான நெட் B1 மாதிரியால் இயக்கப்படுகிறது, இது இலகுரக இன்னும் சக்திவாய்ந்த சுருக்க நரம்பியல் நெட்வொர்க் அதன் உகந்த செயல்திறன் மற்றும் துல்லியத்திற்காக அறியப்படுகிறது. பழப் படங்களில் நுட்பமான காட்சி குறிப்புகளைக் கண்டறிய மாதிரி பயிற்சி அளிக்கப்பட்டது, இது புத்துணர்ச்சியிலிருந்து சிதைவிலிருந்து வேறுபடுத்துகிறது, இது விவசாய மற்றும் சில்லறை சூழல்களில் தரக் கட்டுப்பாட்டுப் பணிகளுக்கு ஏற்றது.

முன்னறிவிப்பு துல்லியம் மற்றும் பொதுமைப்படுத்தலை மேம்படுத்த, குறிப்பாக சுழற்றப்பட்ட அல்லது சற்று மாறுபட்ட உள்ளீட்டுப் படங்களைக் கையாளும் போது, இந்த அமைப்பு சோதனை நேர விரிவாக்கத்தை (TTA) உள்ளடக்கியது. இந்த நுட்பம் அனுமான கட்டத்தில் சுழற்சி போன்ற சீரற்ற பட மாற்றங்களைப் பயன்படுத்துகிறது மற்றும் மிகவும் நம்பகமான வெளியீட்டை உருவாக்க பல முன்னறிவிப்பு முடிவுகளை சராசரியாகக் கொண்டுள்ளது. இந்த விரிவாக்கம் மாறுபட்ட விளக்குகள், கோணங்கள் மற்றும் பின்னணி மாறுபாடுகள் ஆகியவற்றால் ஏற்படும் சவால்களைத் தணிக்க உதவுகிறது, அவை படத் தரவில் பொதுவானவை. மாதிரி அதன் முடிவெடுக்கும் செயல்முறையில் அதிக நுண்ணறிவை வழங்க சிறந்த முன்னறிவிக்கப்பட்ட வகுப்பை நம்பிக்கை மதிப்பெண்ணுடன் வெளியிடுகிறது மேலும் முதல் மூன்று கணிப்புகளை பட்டியலிடுகிறது. பயன்பாட்டு இடைமுகம் ஸ்ட்ரீம் லிட் பயன்படுத்தி கட்டப்பட்டுள்ளது, இது சுத்தமான மற்றும் உள்ளுணர்வு பயனர் அனுபவத்தை வழங்குகிறது. பயனர்கள் JPG அல்லது பதிவேற்றலாம் மற்றும் உடனடி வகைப்பாடு முடிவுகளைப் பெறலாம். கணினி முன்னேற்றப் பட்டைகள் மற்றும் பயனுள்ள காட்சி குறிப்புகளையும் உள்ளடக்கியது. மேம்பட்ட நரம்பியல் கட்டமைப்பு மற்றும் TTA அடிப்படையிலான வலிமை ஆகியவற்றுடன், இந்த வகைப்படுத்தி உணவு தொழில்நுட்பம் மற்றும் விவசாய ஆட்டோமேஷன் துறையில் ஆழமான கற்றலின் நடைமுறை பயன்பாட்டை நிரூபிக்கிறது.

CHAPTER 1

INTRODUCTION

In today's fast-paced world, ensuring the quality and freshness of fruits is essential for both consumers and suppliers. Manual inspection of fruits can be time-consuming, inconsistent, and prone to human error, especially when dealing with large volumes. With the advancement of deep learning technologies, automated image-based fruit classification has emerged as a reliable solution to this problem. This project presents an intelligent fruit classification system capable of distinguishing between fresh and rotten fruits using a Convolutional Neural Network (CNN) model based on the EfficientNetB1 architecture. The system classifies images into six categories—fresh and rotten apples, bananas, and oranges—by analyzing their visual characteristics. To improve accuracy and stability, the model incorporates Test-Time Augmentation (TTA), which generates predictions from multiple augmented versions of the input image and averages the results. The application is built using Stream lit, providing an easy-to-use interface where users can upload fruit images and receive classification with confidence scores. This solution demonstrates the potential of AI in enhancing food quality control and streamlining inspection processes in agricultural and retail settings.

1.1 FRUIT CLASSIFICATION

Fruit classification refers to the process of categorizing different types of fruits based on their visual characteristics, such as color, texture, shape, and size. This task is essential in various industries, including agriculture, food processing, and retail, where determining the freshness and quality of fruits is vital for consumer satisfaction and operational efficiency. In the context of machine learning and artificial intelligence, fruit classification typically involves training a deep learning model, such as a Convolutional Neural Network (CNN), on a labeled dataset of fruit images to recognize patterns and make predictions. The model can classify fruits into different categories, such as fresh or rotten, and can be extended to multiple fruit types, like apples, bananas, and oranges. Automated fruit classification systems reduce human labor, minimize errors, and improve speed and accuracy in sorting fruits, making them highly beneficial in large-scale production and distribution settings.

1.2 EFFICIENT NET B1

Efficient Net B1 is a deep learning model architecture that belongs to the Efficient Net family, which is designed to optimize accuracy while maintaining computational efficiency. It is based on a compound scaling method that uniformly scales all dimensions of the network—depth, width, and resolution—leading to a balanced and efficient model. EfficientNetB1 is one of the smaller variants in the family, offering a good trade-off between performance and resource consumption. It utilizes a novel building block called "MB Conv," which incorporates depth wise separable convolutions to reduce computational cost while preserving model accuracy. The model's performance is state-of-the-art, especially when applied to image classification tasks, making it a popular choice for practical applications like object detection and classification. In this system, EfficientNetB1 is leveraged for fruit classification, providing high accuracy with relatively low computational overhead, which allows it to process images efficiently in applications, even on devices with limited resources.

1.3 Test-Time Augmentation (TTA)

Test-Time Augmentation (TTA) is a technique used to improve the accuracy and robustness of machine learning models during inference by generating multiple augmented versions of the input data at the time of testing. Unlike traditional data augmentation, which is applied during the training phase to increase the diversity of the training data, TTA applies transformations such as rotation, scaling, flipping, and cropping to the test data at the moment of prediction. These augmented versions are then passed through the trained model, and their outputs are averaged or combined to provide a more reliable and stable prediction. This method helps reduce the impact of small variations or noise in the input data, leading to more consistent results, particularly when images may vary in orientation, lighting, or other conditions. In fruit classification tasks, TTA enhances the model's ability to accurately predict fruit types and freshness by considering a range of possible representations of the input image, ultimately improving the confidence and precision of the predictions.

1.4 IMAGE PROCESSING

Image processing refers to the technique of manipulating and analyzing digital images using various algorithms to extract useful information or enhance image quality. It is a critical step in many computer vision tasks, including object detection, classification, and recognition. In the context of fruit classification, image processing involves several stages such as resizing, normalization, color space transformation, and augmentation to ensure that the input images are suitable for model input and that the model can generalize well to new, unseen data. This process may include resizing images to a fixed size, converting them to a specific color format (e.g., RGB), and applying normalization to scale pixel values for improved model performance. Furthermore, techniques like Test-Time Augmentation (TTA) may be used to create augmented versions of images, introducing slight variations like rotation, flipping, and zooming, which helps the model handle variations in image input. Overall, image processing serves as a foundational step in preparing raw images for machine learning models, ensuring they are clean, standardized, and optimized for accurate predictions.

1.5 OBJECTIVES

- To build a machine learning model that can accurately classify fruits as either fresh or rotten, focusing on popular fruits like apples, bananas, and oranges.
- To enhance the model's robustness by applying Test-Time Augmentation techniques, ensuring that the system can handle variations in image quality, lighting, and orientation during inference.
- To design an intuitive and user-friendly interface using Streamlit, allowing users to easily upload fruit images and receive predictions with confidence levels.
- To utilize the EfficientNetB1 architecture for deep learning, ensuring that the system delivers high accuracy while maintaining computational efficiency, making it suitable for practical deployment on devices with limited resources.
- To generate not only the predicted class (fresh or rotten) but also the top three predictions and their corresponding confidence scores, offering users insight into the model's decision-making process and improving trust in the results.

CHAPTER 2

2. LITERATURE REVIEW

2.1 FRU VEG_ MULTI NET: A HYBRID DEEP LEARNING-ENABLED IOT SYSTEM FOR FRESH FRUIT AND VEGETABLE IDENTIFICATION

Reazul Islam et.al has proposed in this system The automatic identification of fresh vegetables and fruits is imperative to streamline agricultural processes, ensuring rapid and accurate assessment of produce quality, reducing economic pressure, and addressing societal needs, particularly for visually impaired individuals. This research presents a pioneering approach for fresh fruit and vegetable identification through IoT and a hybrid deep learning model, combining EfficientNetB7 and ResNet50 architectures. The proposed hybrid model demonstrates remarkable accuracy, achieving 99.92% and 95.93% precision on dataset1 and dataset2, respectively. The study encompasses a comprehensive evaluation of four initial models: EfficientNetB7, VGG16, ResNet50, and VGG19.

The hybrid model, which combines the best of these, performed better than the others. In addition to its high accuracy, the system achieved an average response time of 1.201 s, highlighting its efficiency in processing and decision-making. Considering these challenges in the agricultural industry, the research extends to fruit and vegetable classification, offering applications in self-service fruit or vegetable purchasing, production lines, and smart agriculture. Additionally, the societal impact is considered, with the development of technology aiding the visually impaired in assessing produce freshness. Furthermore, we developed a useful web application that categorizes fruits and vegetables and links to a detailed database offering important information about the recognized produce

The accurate identification and classification of fresh fruits and vegetables play a pivotal role in ensuring the production of high-quality raw materials for the global market and the health sector. Accurate classification and gradation of fruit and vegetable freshness is critical for the agricultural industry, especially in delivering the highest- quality produce to consumers. Numerous illness outbreaks can be traced back to unhealthy fruits and vegetables [1]. The significance of fruit safety in the context of the global economy further underscores the importance of developing advanced technologies [2,3] to address challenges in the agricultural sector. There is growing economic pressure on the agricultural industry [4,5] with an increasing prevalence of infections affecting fruits and vegetables.

Manual sorting of various types of fruits to assess freshness is time-consuming and prone to errors. Automatic classification approaches, such as the hybrid deep learning model proposed in this research, offer a promising solution to expedite the assessment of fruit and vegetable quality. Our study presents a novel approach to fresh fruit and vegetable identification leveraging IoT technology and a hybrid deep learning model. By utilizing an ESP32 webcam for image capture and the SMQTT protocol for data transmission to the cloud, we established a seamless framework for real-time assessment of produce freshness. Our hybrid model, integrating EfficientNetB7 and ResNet50 architectures, demonstrated remarkable accuracy, achieving 99.92% and 95.93% accuracy rates for dataset1 and dataset2, respectively.

2.2 RECENT ADVANCEMENTS IN FRUIT DETECTION AND CLASSIFICATION USING DEEP LEARNING TECHNIQUES

Recent advancements in fruit detection and classification using deep learning techniques have revolutionized the way the agricultural and food industries manage fruit quality. Convolutional Neural Networks (CNNs), a powerful class of deep learning models, have shown great promise in automatically identifying and classifying fruits based on their appearance, significantly outperforming traditional image processing techniques. These models are trained on large datasets containing high-resolution images of various fruits, enabling them to learn complex patterns and features associated with different fruit types, as well as distinguish between fresh and spoiled fruits.

Among the state-of-the-art CNN architectures, pre-trained models such as VGG16, ResNet, and Inception have proven particularly effective in fruit detection tasks, achieving high accuracy rates when applied to fruit classification problems. In addition to fruit type classification, CNNs are now being used to detect specific signs of spoilage, such as discoloration, mold, bruising, and other visible indicators of rot. This capability allows for more accurate identification of rotten fruits, which would otherwise be challenging to detect using manual inspection methods. Furthermore, deep learning models can be combined with machine learning algorithms like Random Forest or Gaussian Naïve Bayes to predict the shelf life of fruits, providing valuable insights into the time-sensitive nature of their freshness. Incorporating advanced data augmentation techniques and real-time image analysis has further improved these systems' robustness and adaptability to varying conditions. For example, environmental factors such as lighting changes, background noise, and different angles can affect the quality of images used in fruit detection. Deep learning models have been enhanced to mitigate these challenges, offering reliable performance in diverse operational environments. These advancements are making automated fruit quality assessment more accessible, scalable, and cost-effective, ultimately driving down food waste and ensuring that only high-quality, fresh produce reaches consumers. As these systems become more refined, their applications are expanding to large-scale commercial use, supporting smarter, safer, and more efficient food distribution networks.

2.3 FRUIT QUALITY IDENTIFICATION USING IMAGE PROCESSING, MACHINE LEARNING, AND DEEP LEARNING: A REVIEW

The identification of fruit quality plays a crucial role in ensuring the freshness, safety, and overall quality of produce in the agricultural and food industries. Traditional methods of fruit quality assessment often rely on human inspection, which is time-consuming, subjective, and prone to errors. Recent advancements in image processing, machine learning, and deep learning techniques have provided innovative solutions to automate and improve fruit quality identification, enabling faster, more accurate assessments that can be applied on a large scale.

Image processing techniques have laid the foundation for fruit quality identification by extracting essential features from fruit images. These techniques involve preprocessing steps such as color analysis, texture extraction, and shape detection, which help distinguish between fresh and rotten fruits. However, these traditional methods often struggle with complex variations in fruit appearance, especially when spoilage is subtle or occurs under different environmental conditions. To address these limitations, machine learning models, such as Support Vector Machines (SVM), Decision Trees, and k-Nearest Neighbors (k-NN), have been integrated with image processing techniques to enhance the classification accuracy of fruit quality. These models learn from extracted features to distinguish between different fruit types and identify signs of damage, spoilage, or ripeness. Deep learning, particularly Convolutional Neural Networks (CNNs), has brought a paradigm shift in fruit quality identification. CNNs, which are capable of automatically learning relevant features from raw image data, have shown superior performance in handling complex tasks such as classifying fruits based on their quality and detecting defects.

The combination of image processing, machine learning, and deep learning techniques has proven effective in creating robust systems for fruit quality detection. These systems can be deployed in real-time applications, such as automated fruit sorting in warehouses or supermarkets, reducing reliance on manual labor while improving the consistency and accuracy of quality assessments. Furthermore, these technologies are scalable and can handle large volumes of data, making them suitable for large-scale commercial use in the agriculture sector. As research continues to advance, the integration of other emerging technologies, such as Internet of Things (IoT) sensors and cloud computing, is expected to further enhance the capabilities of fruit quality identification systems, enabling more precise monitoring of fruit conditions throughout the entire supply chain. In conclusion, the integration of image processing, machine learning, and deep learning techniques has significantly advanced the field of fruit quality identification. These technologies offer numerous benefits, including increased efficiency, improved accuracy, and enhanced food safety. With continued developments and the integration of new technologies, automated systems for fruit quality detection will continue to play an essential role in the future of agriculture, benefiting both producers and consumers.

2.4 A GENERAL MACHINE LEARNING MODEL FOR ASSESSING FRUIT QUALITY USING DEEP IMAGE FEATURES

Ioannis D. Apostolopoulos et.al has proposed in this system Fruit quality is a critical factor in the produce industry, affecting producers, distributors, consumers, and the economy. High-quality fruits are more appealing, nutritious, and safe, boosting consumer satisfaction and revenue for producers. Artificial intelligence can aid in assessing the quality of fruit using images. This paper presents a general machine learning model for assessing fruit quality using deep image features. This model leverages the learning capabilities of the recent successful networks for image classification called vision transformers (ViT). The ViT model is built and trained with a combination of various fruit datasets and taught to distinguish between good and rotten fruit images based on their visual appearance and not predefined quality attributes. The general model demonstrated impressive results in accurately identifying the quality of various fruits, such as apples (with a 99.50% accuracy), cucumbers (99%), grapes (100%), kakis (99.50%), oranges (99.50%), papayas (98%), peaches (98%), tomatoes (99.50%), and watermelons (98%).

It showed slightly lower performance in identifying guavas (97%), lemons (97%), limes (97.50%), mangoes (97.50%), pears (97%), and pomegranates (97%). Fruit quality refers to a fruit's overall characteristics that determine its desirability, nutritional content, and safety for consumption [1]. It is determined by the fruit's appearance, flavour, texture, nutritional value, and safety [2]. For several reasons, high fruit quality is crucial for the industry, consumers, and the economy. High- quality fruits benefit growers and sellers economically, promote healthy eating habits, reduce healthcare costs, positively impact the environment, ensure food safety, and promote international trade [3].

Promoting high fruit quality requires using sustainable farming practices, implementing food safety regulations, and promoting healthy eating habits [3]. For the industry, fruit quality is critical for market competitiveness and profitability. The produce industry is highly competitive, and consumers are more discerning than ever, demanding high-quality fruits that meet their flavour, appearance, and nutrition expectations. Furthermore, the reputation of producers and distributors depends on the quality of their products [3]. Consumers who are satisfied with the quality of fruits are more likely to become repeat customers and recommend the products to others, which can help to build a strong brand image and increase sales.

AI-based technologies can potentially revolutionise the fruit industry by providing objective and efficient quality assessment. This study introduced a general machine learning model based on vision transformers to assess fruit quality from images. The model outperformed dedicated models trained on single fruit types, except for apples, oranges, and peaches, where both had similar accuracy. Dedicated models were better for specific fruits such as bananas and pomegranates. Overall, a generalised model worked well for most fruit types.

However, dedicated models could improve the accuracy for fruit types with unique features. Fruit quality depends on multiple factors, including appearance, flavour, and nutrition. Appearance can be misleading and affected by various factors. This study has limitations in this regard. Finally, while the 16 fruit types used in this study provide a valid starting point, future research should include a more diverse and extensive range of fruit types to better evaluate the effectiveness of generalised and dedicated models in predicting fruit quality.

2.5 BANANA RIPENESS STAGE IDENTIFICATION: A DEEP LEARNING APPROACH

Banana ripeness stage identification is a critical task in the agricultural industry, as the ripeness of bananas affects their marketability, flavor, and texture. Accurately determining the ripeness stage of bananas is important for both producers and consumers, as it helps in quality control, shelf life prediction, and proper handling during distribution. Traditional methods of ripeness identification often rely on manual inspection, which is subjective and can be inconsistent. As a result, there is a growing interest in automated systems that can objectively and efficiently assess banana ripeness, particularly using deep learning approaches.

Deep learning, especially Convolutional Neural Networks (CNNs), has proven to be an effective method for classifying bananas based on their ripeness stages. CNNs are capable of learning hierarchical features directly from raw image data, making them ideal for fruit ripeness detection tasks. In the case of banana ripeness identification, CNN models are trained on large datasets containing images of bananas at various ripeness stages. The model learns to detect key visual characteristics such as color, texture, and shape, which change as the banana transitions from unripe (green) to fully ripe (yellow or brown). \

These features allow deep learning models to classify bananas into distinct ripeness categories, including unripe, ripe, and overripe stages.The deep learning approach for banana ripeness stage identification involves several key steps. First, a dataset of banana images, typically captured under controlled lighting conditions, is collected. Preprocessing techniques, such as image augmentation and normalization, are applied to enhance the model's ability to generalize across varying environmental factors and fruit appearances. Then, a CNN model is trained to recognize and classify the different ripeness stages. Popular CNN architectures, such as VGG16, ResNet, and Inception, have shown strong performance in similar tasks.

The trained model is evaluated on its ability to classify bananas accurately, and its performance is often measured using metrics such as accuracy, precision, and recall.This approach provides several advantages over traditional methods of ripeness identification. Deep learning models can process large volumes of images in real time, offering scalability and efficiency for commercial applications, such as automated banana sorting in warehouses or supermarkets. Furthermore, these systems are more objective and consistent than manual inspection, as they rely on data-driven insights rather than human judgment. Deep learning-based ripeness detection also allows for non-destructive testing, meaning bananas do not need to be cut or damaged for evaluation, preserving their quality.

Despite the promising results, challenges remain in developing deep learning models for banana ripeness identification. Variability in lighting conditions, camera angles, and backgrounds can affect the quality of the images and, consequently, the performance of the model. Moreover, different banana cultivars may have slightly varying ripening patterns, which requires the model to be adaptable across different varieties.

However, with the continuous advancement of deep learning techniques, including improvements in model architectures and the use of transfer learning, these challenges can be mitigated, leading to more robust and accurate systems for banana ripeness stage identification. In conclusion, deep learning offers a powerful solution for banana ripeness stage identification, automating the process and providing significant benefits in terms of efficiency, accuracy, and scalability. As the agricultural industry continues to embrace technological advancements, deep learning-based systems will play a key role in optimizing fruit quality control and ensuring that only the best produce reaches consumers.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The freshness of fruits is considered to be one of the essential characteristics for consumers in determining their quality, flavor and nutritional value. The primary need for identifying rotten fruits is to ensure that only fresh and high-quality fruits are sold to consumers. The impact of rotten fruits can foster harmful bacteria, molds and other microorganisms that can cause food poisoning and other illnesses to the consumers. The overall purpose of the study is to classify rotten fruits, which can affect the taste, texture, and appearance of other fresh fruits, thereby reducing their shelf life.

The agriculture and food industries are increasingly adopting computer vision technology to detect rotten fruits and forecast their shelf life. Hence, this research work mainly focuses on the Convolutional Neural Network's (CNN) deep learning model, which helps in the classification of rotten fruits. The proposed methodology involves real-time analysis of a dataset of various types of fruits, including apples, bananas, oranges, papayas and guavas. Similarly, machine learning models such as Gaussian Naïve Bayes (GNB) and random forest are used to predict the fruit's shelf life.

The results obtained from the various pre-trained models for rotten fruit detection are analysed based on an accuracy score to determine the best model. In comparison to other pretrained models, the visual geometry group16 (VGG16) obtained a higher accuracy score of 95%. Likewise, the random forest model delivers a better accuracy score of 88% when compared with GNB in forecasting the fruit's shelf life. By developing an accurate classification model, only fresh and safe fruits reach consumers, reducing the risks associated with contaminated produce. Thereby, the proposed approach will have a significant impact on the food industry for efficient fruit distribution and also benefit customers to purchase fresh fruits.

3.1.1 DRAWBACKS

- The accuracy of the CNN model and other machine learning models is highly dependent on the quality and variety of the training data. If the dataset doesn't include sufficient representations of all fruit types and spoilage stages, the model may struggle with generalization, leading to misclassification.
- The current system primarily focuses on fruits like apples, bananas, oranges, papayas, and guavas. This narrow focus may limit the system's applicability in scenarios where a broader range of fruit types is involved, requiring further adaptation for diverse fruit categories.
- Deep learning models like CNN, particularly pre-trained models like VGG16, require significant computational resources for training and real-time inference. This can increase costs and may not be feasible for smaller operations without access to powerful hardware.
- The system might struggle with variations in environmental factors like lighting, background, or fruit position, which can impact the image quality and affect the model's performance. Such challenges may lead to incorrect classifications, especially in less controlled environments.
- While CNNs, especially pre-trained models, are powerful, there is a risk of overfitting if the model is trained on limited or highly specific data. This could result in poor generalization to unseen data or new fruit types, reducing the model's effectiveness in applications.

3.2 PROPOSED SYSTEM

The proposed system is a deep learning-based image classification model designed to identify the freshness status of fruits using EfficientNetB1 architecture. The system accepts images of apples, bananas, and oranges, and classifies them into one of six categories: fresh apple, fresh banana, fresh orange, rotten apple, rotten banana, and rotten orange. To ensure robustness against variations in image orientation, lighting, and background, the system incorporates Test-Time Augmentation (TTA), where the input image is randomly rotated multiple times, and predictions from these augmented images are averaged to provide a more stable and confident output. This technique enhances the model's ability to generalize across diverse and potentially noisy input data, which is common in applications.

The classification model is deployed through an interactive web interface developed using Streamlit, which enables users to upload fruit images and receive predictions. Upon uploading an image, the system preprocesses it, performs TTA, and uses the trained EfficientNetB1 model to predict the freshness category. It then displays the most probable class along with its confidence level, as well as the top three class predictions for better interpretability. The integration of deep learning with a user-friendly front end allows for practical and accessible use in fields such as food quality inspection, inventory management, and supply chain monitoring.

3.2.1 ADVANTAGES

- The system provides accurate and reliable predictions by leveraging a deep learning model, ensuring that fruits are classified correctly based on their visual features.
- By incorporating Test-Time Augmentation, the model becomes more robust, improving its performance under various conditions, such as variations in lighting, orientation, and image quality.
- The user-friendly interface allows anyone, even with minimal technical knowledge, to easily upload images and receive instant results, making it accessible for a wide range of users.

- The EfficientNetB1 architecture used in the model optimizes computational efficiency, delivering high accuracy while minimizing resource consumption, which is ideal for both cloud-based and edge-device applications.
- The inclusion of confidence scores and top-three predictions provides transparency, helping users understand the model's decision-making process and enhancing trust in the results.

3.3 FEASIBILITY STUDY

Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility
- Economical Feasibility

3.3.1 TECHNICAL FEASIBILITY

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at DB2 Database. Thus it provides an easy access to the users. The database's purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

Therefore, it provides the technical guarantee of accuracy, reliability and security. The software and hard requirements for the development of this project are not many and are already available in-house at NIC or are available as free as open source. The work for the project is done with the current equipment and existing software technology. Necessary bandwidth exists for providing a fast feedback to the users irrespective of the number of users using the system.

3.3.2 OPERATIONAL FEASIBILITY

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization's operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

- Is there sufficient support for the management from the users?
- Will the system be used and work properly if it is being developed and implemented?
- Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

3.3.3 ECONOMIC FEASIBILITY

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS:

- Processor Type : AMD RYZEN 7
- Speed : 4.40GHZ
- RAM :16 GB RAM
- Hard disk : 1 TB
- Keyboard : 101/102 Standard Keys
- Mouse : Optical Mouse

4.2 SOFTWARE SPECIFICATION

- Operating System : Windows 10
- Front End : Jupyter Notebook/ Anaconda tool
- Coding Language : Python

CHAPTER 5

SOFTWARE DESCRIPTION

5.1 FRONT END: FEATURES OF SOFTWARE

FRONT END: JUPYTER NOTEBOOK

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more. The software requirement specification is created at the end of the analysis task. The function and performance allocated to software as part of system engineering are developed by establishing a complete information report as functional representation, a representation of system behaviour, an indication of performance requirements and design constraints, appropriate validation criteria.

FEATURES OF JUPYTER NOTEBOOK

- In-browser editing for code, with automatic syntax highlighting, indentation, and tab completion/introspection.

- The ability to execute code from the browser, with the results of computations attached to the code which generated them.

- Displaying the result of computation using rich media representations, such as HTML, LaTeX, PNG, SVG, etc.

- For example, publication-quality figures rendered by the matplotlib library, can be included inline.

- In-browser editing for rich text using the Markdown markup language, which can provide commentary for the code, is not limited to plain text.

- The ability to easily include mathematical notation within markdown cells using LaTeX, and rendered natively by MathJax.

NOTEBOOK DOCUMENTS

Notebook documents contains the inputs and outputs of a interactive session as well as additional text that accompanies the code but is not meant for execution.

In this way, notebook files can serve as a complete computational record of a session, interleaving executable code with explanatory text, mathematics, and rich representations of resulting objects. These documents are internally JSON files and are saved with the .ipynb extension. Since JSON is a plain text format, they can be version-controlled and shared with colleagues. Notebooks may be exported to a range of static formats, including HTML (for example, for blog posts), Restructured Text, LaTeX, PDF, and slide shows, via the nb convert command.

Furthermore, any .ipynb notebook document available from a public URL can be shared via the Jupiter Notebook Viewer (nb viewer). This service loads the notebook document from the URL and renders it as a static web page. The results may thus be shared with a colleague, or as a public blog post, without other users needing to install the Jupyter notebook themselves. In effect, nb viewer is simply nb convert as a web service, so you can do your own static conversions with nb convert, without relying on nb viewer.

PYTHON DEFINITION

Python is a high-level programming language designed to be easy to read and simple to implement. It is open source, which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to Java and .NET virtual machines.

Python is considered a scripting language, like Ruby or Perl and is often used for creating Web applications and dynamic Web content. It is also supported by a number of 2D and 3D imaging programs, enabling users to create custom plug-ins and extensions with Python. Examples of applications that support a Python API include GIMP, Inkscape, Blender, and Autodesk Maya.

Scripts written in Python (.PY files) can be parsed and run immediately. They can also be saved as a compiled programs (.PYC files), which are often used as programming modules that can be referenced by other Python programs.

PYTHON FEATURES

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

1) Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

2) Expressive Language

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type print("Hello World"). It will take only one line to execute, while Java or C takes multiple lines.

3) Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4) Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

5) Free and Open Source

Python is freely available for everyone. It is freely available on its official website www.python.org. It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

6) Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

7) Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

8) Large Standard Library

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

9) GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

10) Integrated

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C,C++ Java. It makes easy to debug the code.

11) Embeddable

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

12) Dynamic Memory Allocation

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 15 to x, then we don't need to write int x = 15. Just write x = 15.

ANACONDA

Anaconda Cloud is a package management service by Anaconda. Cloud makes it easy to find, access, store and share public notebooks, environments, and conda and PyPI packages. Cloud also makes it easy to stay current with updates made to the packages and environments you are using. Cloud hosts hundreds of useful Python packages, notebooks, projects and environments for a wide variety of applications. You do not need to log in, or even to have a Cloud account, to search for public packages, download and install them.

You can build new conda packages using conda-build, then upload the packages to Cloud to quickly share with others or access yourself from anywhere. The Anaconda Cloud command line interface (CLI), anaconda-client, allows you to manage your account - including authentication, tokens, upload, download, remove and search.Connect to and manage your Anaconda Cloud account. Upload packages you have created. Generate access tokens to allow access to private packages.

For developers, Cloud is designed to make software development, release and maintenance easy by providing broad package management support. Cloud allows for free public package hosting, as well as package channels, providing a flexible and scalable service for groups and organizations of all sizes.

APPLICATIONS PROVIDED IN ANACONDA DISTRIBUTION

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

- JupyterLab
- Jupyter Notebook
- Qt Console
- Spyder
- Glueviz
- Orange3
- RStudio
- Visual Studio Code

JupyterLab: This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Jupyter Notebook: This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.

Qt Console: It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.

Spyder: Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.

VS Code: It is a streamlined code editor with support for development operations like debugging, task running and version control.

Glueviz: This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.

Orange 3: It is a component-based data mining framework. This can be used for data visualization and data analysis. The workflows in Orange 3 are very interactive and provide a large toolbox.

Rstudio: It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks

CHAPTER 6

PROJECT DESCRIPTION

6.1 PROBLEM DEFINITION

The problem at hand is the identification and classification of rotten fruits to ensure that only fresh and high-quality produce is sold to consumers. Rotten fruits pose significant risks, not only due to their degraded taste, texture, and appearance but also because of the potential growth of harmful bacteria, molds, and microorganisms that can cause foodborne illnesses. In the food and agriculture industries, manual inspection of fruits is time-consuming, prone to human error, and often inefficient. To address this issue, the need arises for an automated, reliable, and efficient system that can detect rotten fruits in real time, classify them accurately, and forecast their remaining shelf life. This system is crucial for ensuring consumer safety, minimizing food waste, and enhancing the overall efficiency of fruit distribution processes. The challenge lies in developing a model that can reliably distinguish between fresh and rotten fruits across various types, handling variability and offering scalable solutions for large-scale deployment.

6.2 MODULE DESCRIPTION

6.2.1. IMAGE UPLOAD

This module serves as the entry point for users to interact with the system. It enables users to upload fruit images in supported formats such as JPG, JPEG, or PNG. The uploaded image is temporarily stored and displayed on the interface for confirmation. This module ensures that the system receives valid and usable image inputs while maintaining a user-friendly experience. It acts as a crucial bridge between the user and the backend processing pipeline.

6.2.2. PREPROCESSING

Once an image is uploaded, it is passed to the preprocessing module, which standardizes the image before feeding it into the classification model. This includes converting the image to RGB, resizing it to a fixed dimension (240x240 pixels), and normalizing pixel values to a scale of 0 to 1. These steps are essential for maintaining consistency across various input images and ensuring compatibility with the trained EfficientNetB1 model. The preprocessing also helps reduce noise and enhances the model's ability to make accurate predictions.

6.2.3. CLASSIFICATION

At the core of the system lies the classification module, which leverages a pre-trained EfficientNetB1 model. This deep learning model has been trained to distinguish between six classes of fruits—fresh and rotten variants of apples, bananas, and oranges. The module receives the preprocessed image and generates predictions based on learned features. It identifies subtle differences in texture, color, and shape to accurately classify the image into its respective category.

6.2.4. TEST-TIME AUGMENTATION (TTA)

To improve the robustness and accuracy of predictions, the system incorporates Test-Time Augmentation. This module applies a series of random rotations to the input image and performs multiple inference passes using the augmented versions. The results of these passes are then averaged to produce a final prediction. TTA helps the model handle variations such as different angles or slight distortions in user-uploaded images, thus enhancing its reliability.

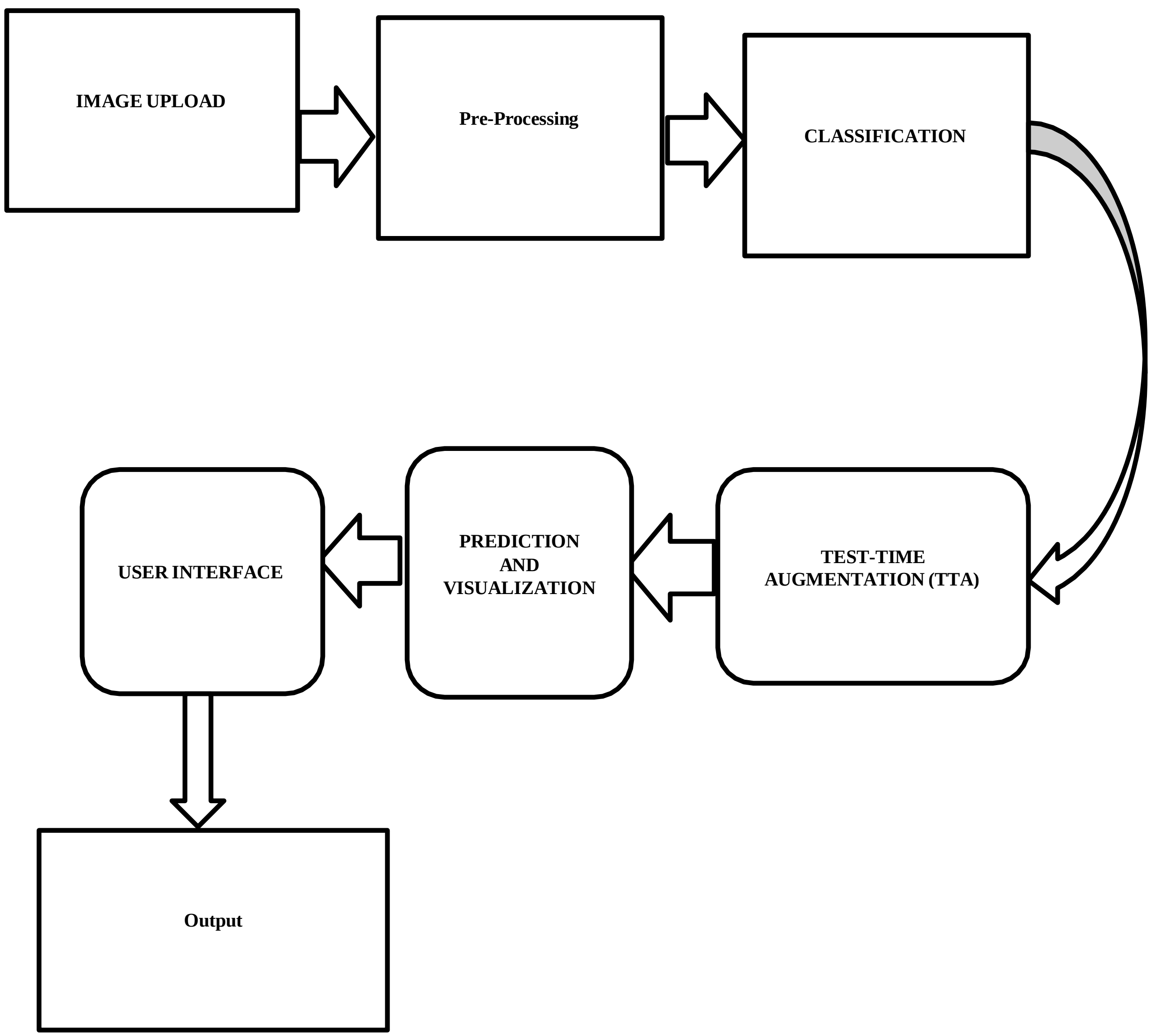
6.2.5. PREDICTION AND VISUALIZATION

After the classification is complete, the prediction and visualization module displays the results in an informative manner. It shows the predicted class label along with a confidence percentage, helping users understand the certainty of the model's output. Additionally, it presents the top three class predictions with progress bars, offering deeper insight into the model's reasoning. This enhances transparency and user trust in the system's performance.

6.2.6. USER INTERFACE

The user interface module is built using Streamlit, offering a clean, responsive, and interactive platform for end-users. It integrates all other modules into a seamless workflow, guiding users from image upload to result interpretation. The interface includes visual feedback, loading indicators during processing, and aesthetic design elements like emojis and headers to make the experience engaging and intuitive. It ensures that even users with no technical background can interact with the system effortlessly.

6.3 SYSTEM FLOW DIAGRAM



6.4 INPUT DESIGN

The input design of the fruit classification system focuses on creating a simple, user-friendly interface that facilitates accurate and efficient image submission. Users are prompted to upload an image of a fruit through a file uploader component that accepts common image formats such as JPG, JPEG, and PNG. The uploaded image is instantly displayed on the interface, allowing users to verify the correctness and clarity of the image before classification. To ensure compatibility with the backend model, the input design automatically handles preprocessing tasks such as resizing the image to 240x240 pixels and converting it to RGB format. This standardization step is crucial for maintaining uniformity and ensuring optimal model performance. Additionally, the input design is robust enough to accept images captured from various devices and angles, as the system employs Test-Time Augmentation to account for variations. Overall, the input design ensures a smooth and intuitive data entry process while maintaining the technical requirements needed for accurate prediction.

6.5 OUTPUT DESIGN

The output design of the fruit freshness classification system is structured to present results in a clear, informative, and visually engaging manner. Once the image has been processed and classified, the system displays the predicted class label—such as fresh apple or rotten banana—along with the confidence level as a percentage, giving users a direct understanding of the model's certainty. To enhance transparency, the system also showcases the top three predicted classes, each accompanied by a progress bar that visually represents the probability score. This multi-layered output helps users assess alternative classifications in cases where confidence is distributed among similar classes. The design ensures that even non-technical users can interpret the results easily, with helpful labels, icons, and formatting elements like bold text and color indicators. Overall, the output design is both informative and user-centric, focusing on readability, clarity, and user engagement.

CHAPTER 7

SYSTEM TESTING AND IMPLEMENTATION

7.1 SYSTEM TESTING

System testing plays a vital role in ensuring the reliability, accuracy, and usability of the fruit freshness classification system. The entire system was subjected to rigorous functional and non- functional testing to validate its performance under various conditions. Functional testing was carried out to verify that each module—such as image uploading, preprocessing, classification, Test-Time Augmentation, and result visualization—operated correctly and interacted seamlessly with one another. Test cases included uploading images of different formats, resolutions, orientations, and lighting conditions to assess the system's robustness. Accuracy testing was conducted using a diverse set of fruit images to evaluate the model's predictive performance across all six classes. The system was also tested for responsiveness, including how quickly it handled image uploads, processed data, and delivered results. Furthermore, the user interface underwent usability testing to ensure it remained intuitive and accessible to users with minimal technical knowledge. Overall, the testing process confirmed that the system functions as intended and is capable of delivering reliable and user-friendly fruit classification results.

7.2 SYSTEM IMPLEMENTATION

The implementation of the fruit freshness classification system was carried out using a combination of deep learning and interactive web technologies to ensure both accuracy and accessibility. The core model is based on EfficientNetB1, a high-performing convolutional neural network that was pre-trained and fine-tuned using a labeled dataset containing images of fresh and rotten fruits. This model was saved in H5 format and integrated into the application using TensorFlow. The front-end interface was developed using Streamlit, allowing for the rapid deployment of a clean, responsive web application. During implementation, modules were carefully structured to manage image upload, preprocessing, prediction with Test-Time Augmentation, and result display. The system was hosted in a way that allowed seamless interaction with the trained model, enabling inference.

Each component was tested during the integration phase to ensure smooth data flow and functionality. Overall, the implementation focused on building a scalable, modular system that delivers accurate classifications while offering a user-friendly interface suitable for practical deployment.

CHAPTER 8

CONCLUSION & FUTURE WORK

In conclusion, the fruit freshness classification system effectively combines deep learning and image processing techniques to offer an automated solution for distinguishing between fresh and rotten fruits. By utilizing the EfficientNetB1 model with Test-Time Augmentation (TTA), the system ensures high accuracy and robustness, even under varied and challenging conditions. The user-friendly interface built with Stream lit allows for easy interaction, making it accessible to users from different backgrounds. Additionally, the model's efficiency in terms of computational resources enables its deployment on a wide range of devices, from high-performance servers to more limited edge devices. The system's transparent output, including confidence scores and top predictions, further enhances its reliability and trustworthiness. Overall, this approach not only streamlines fruit quality inspection but also paves the way for future advancements in automated agricultural and retail applications.

FUTURE WORK

For future work, the fruit freshness classification system can be further enhanced by expanding the dataset to include a wider variety of fruits and more diverse environmental conditions, such as different lighting, backgrounds, and angles. This would help improve the model's generalization and robustness. Additionally, integrating other advanced deep learning techniques, such as transfer learning or reinforcement learning, could further boost performance, especially in handling edge cases like partially damaged fruits. Another promising avenue would be the development of a mobile application that allows users to upload images directly from their smartphones for on-the-go fruit inspection. The system could also be integrated with monitoring systems in agriculture or retail to automate quality control and improve supply chain efficiency. Furthermore, incorporating multi-class classification for various levels of fruit ripeness and incorporating additional features, such as texture analysis through specialized sensors, could further refine the predictions. Overall, continued advancements in AI and hardware optimization hold great potential for expanding the applicability and performance of such systems.

CHAPTER 9

APPENDICES

9.1 SOURCE CODE

```
import os
import cv2
import numpy as np
from tqdm import tqdm

def load_fresh_rotten_data():
    X = []
    Y = []
    base_dir = './input/dataset/train'

    # List of fresh fruit folders
    fresh_folders = ['freshapples', 'freshbanana', 'freshoranges']
    # List of rotten fruit folders
    rotten_folders = ['rottenapples', 'rottenbanana', 'rottenoranges']

    # Process fresh fruits (label = 0)
    for fresh_folder in fresh_folders:
        path = os.path.join(base_dir, fresh_folder)
        for img_name in tqdm(os.listdir(path)):
            img = cv2.imread(os.path.join(path, img_name))
            img = cv2.resize(img, (100, 100))
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            X.append(img)
            Y.append(0)

    for rotten_folder in rotten_folders:
```

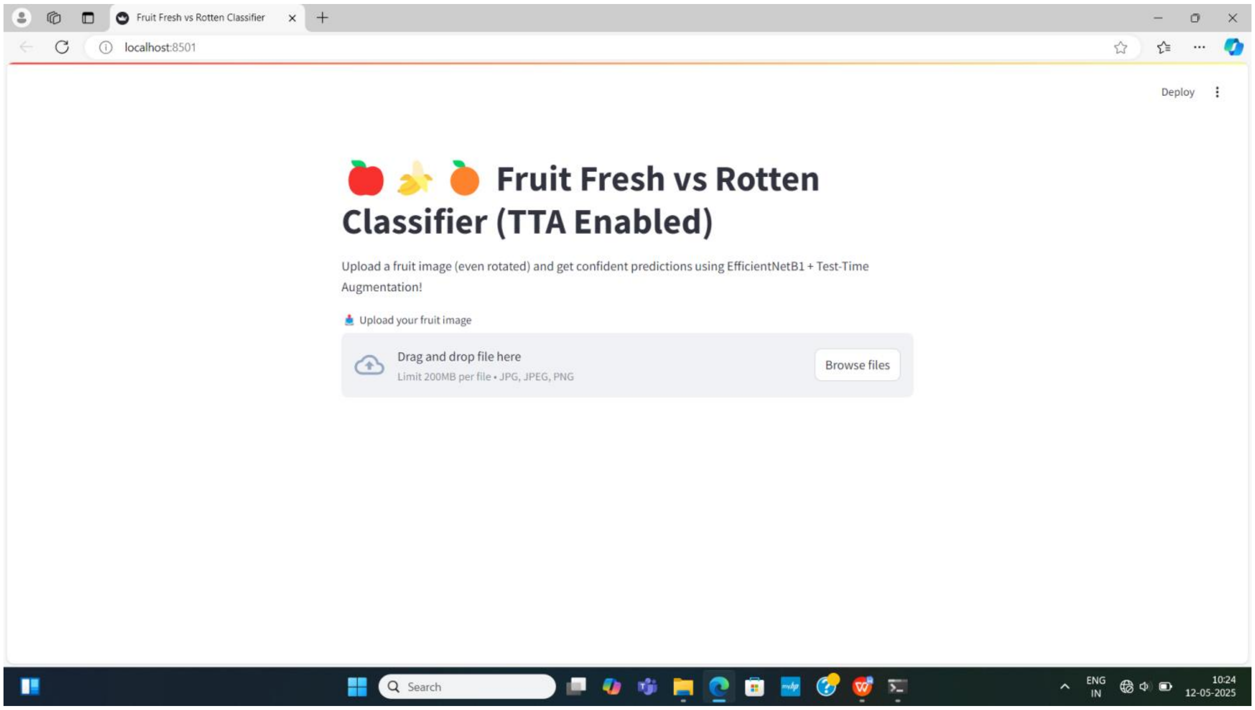
```
        path = os.path.join(base_dir, rotten_folder)  # Path to each rotten fruit folder
        for img_name in tqdm(os.listdir(path)):
            img = cv2.imread(os.path.join(path, img_name))
            img = cv2.resize(img, (100, 100))
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            X.append(img)
            Y.append(1)

    return np.array(X), np.array(Y)

X, Y = load_fresh_rotten_data()
X = X / 255.0
  import matplotlib.pyplot as plt  # For displaying images

# Function to plot training and validation accuracy/loss
def plot_training_history(history):
    # Plot accuracy
    plt.figure(figsize=(12, 4))

    plt.subplot(1, 2, 1)
    plt.plot(history.history['accuracy'], label='Training Accuracy')
    plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
    plt.title('Training and Validation Accuracy')
    plt.xlabel('Epochs')
    plt.ylabel('Accuracy')
    plt.legend()

    # Plot loss
    plt.subplot(1, 2, 2)
    plt.plot(history.history['loss'], label='Training Loss')
    plt.plot(history.history['val_loss'], label='Validation Loss')
```

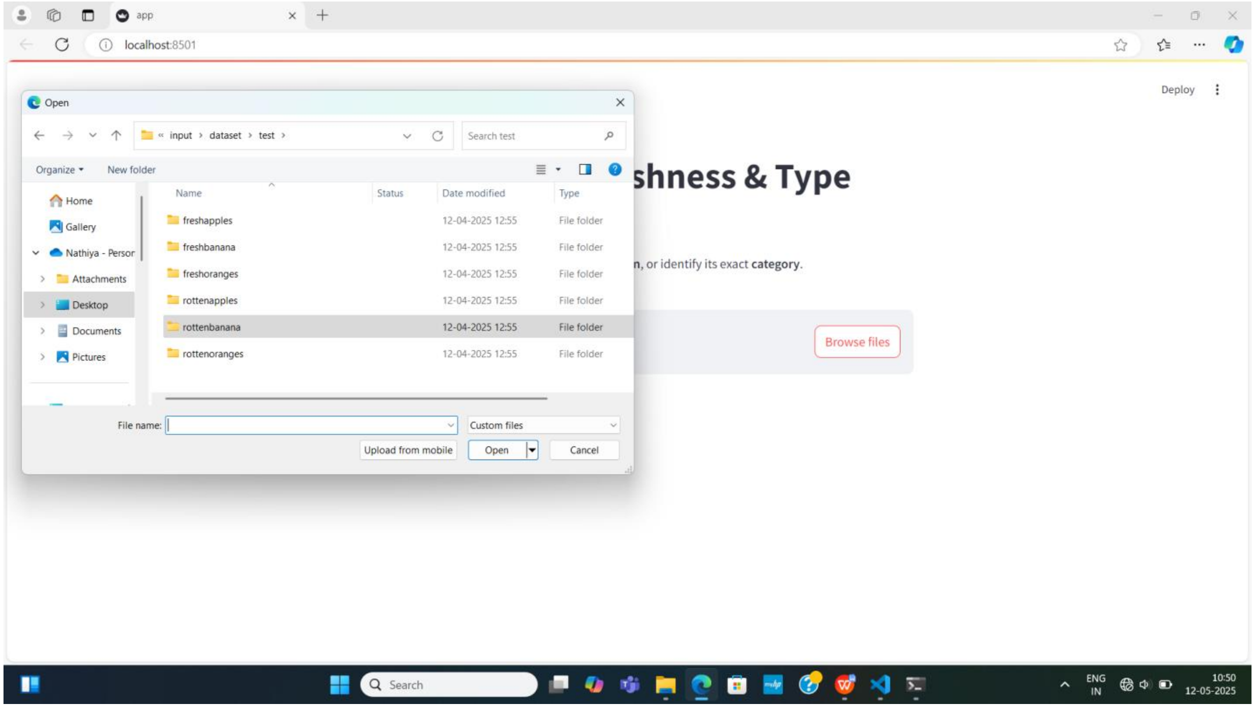
```python
    plt.title('Training and Validation Loss')
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.legend()

    plt.show()

# Plot training and validation history
plot_training_history(history)
import matplotlib.pyplot as plt  # For displaying images

from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Get predictions for the validation set
Y_pred = model_fresh_rotten.predict(X_val)
Y_pred = (Y_pred > 0.5).astype(int)  # Convert probabilities to binary predictions

# Compute confusion matrix
cm = confusion_matrix(Y_val, Y_pred)

# Display confusion matrix
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Fresh', 'Rotten'])
disp.plot(cmap=plt.cm.Blues)
plt.show()
from tensorflow.keras.preprocessing import image

# Function to load and preprocess the image for prediction
def load_image_for_prediction(img_path):
    img = image.load_img(img_path, target_size=(100, 100))  # Load and resize the image
    img_array = image.img_to_array(img)  # Convert image to array
    img_array = np.expand_dims(img_array, axis=0)  # Expand dimensions for model input
```

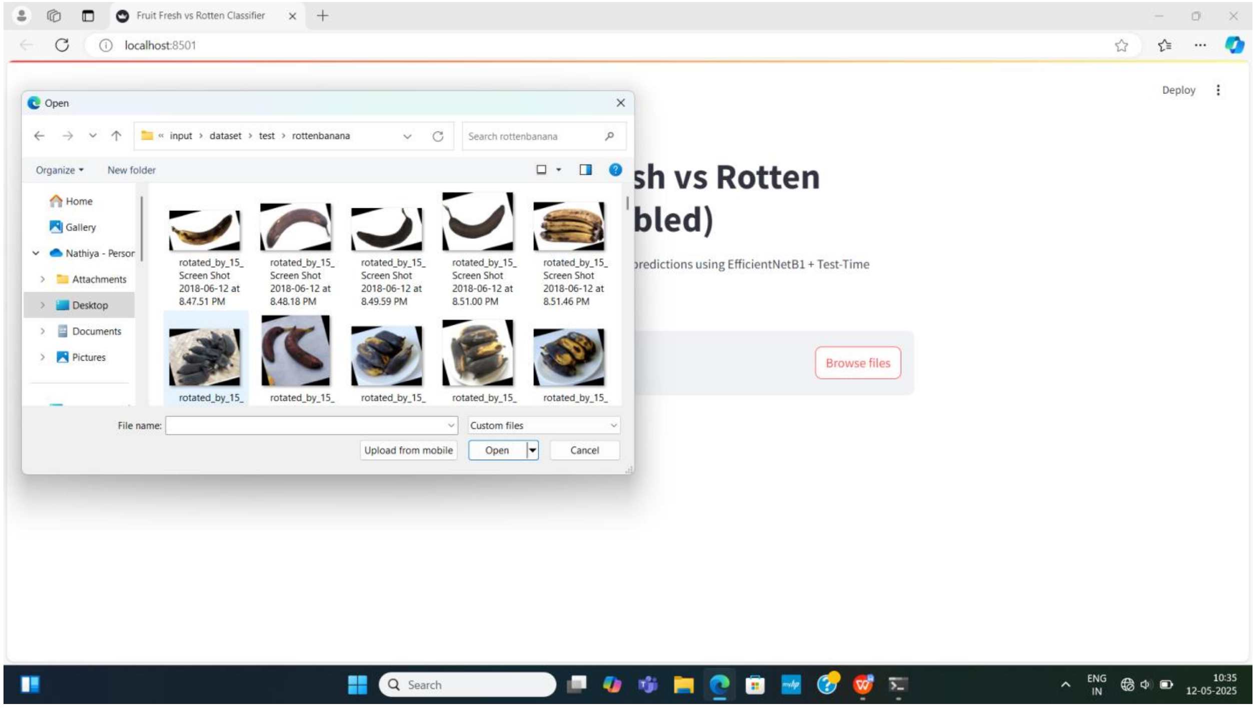
```
    img_array = img_array / 255.0  # Normalize the image

    return img_array

# Function to predict if the fruit is fresh or rotten
def predict_fresh_or_rotten(model, img_path):
    img_array = load_image_for_prediction(img_path)
    prediction = model.predict(img_array)

    if prediction > 0.5:
        return 'Rotten'
    else:
        return 'Fresh'

# Example usage:
img_path = './input/dataset/train/rottenapples/Screen Shot 2018-06-07 at 2.15.20 PM.png'
result = predict_fresh_or_rotten(model_fresh_rotten, img_path)
print(f"The fruit is: {result}")
def load_test_data():
    X_test = []
    Y_test = []
    test_dir = './input/dataset/test'

    # Define the folders for fresh and rotten test images
    fresh_folders = ['freshapples', 'freshbanana', 'freshoranges']
    rotten_folders = ['rottenapples', 'rottenbanana', 'rottenoranges']

    # Load fresh images (label = 0)
    for fresh_folder in fresh_folders:
        path = os.path.join(test_dir, fresh_folder)
        for img_name in tqdm(os.listdir(path)):
```

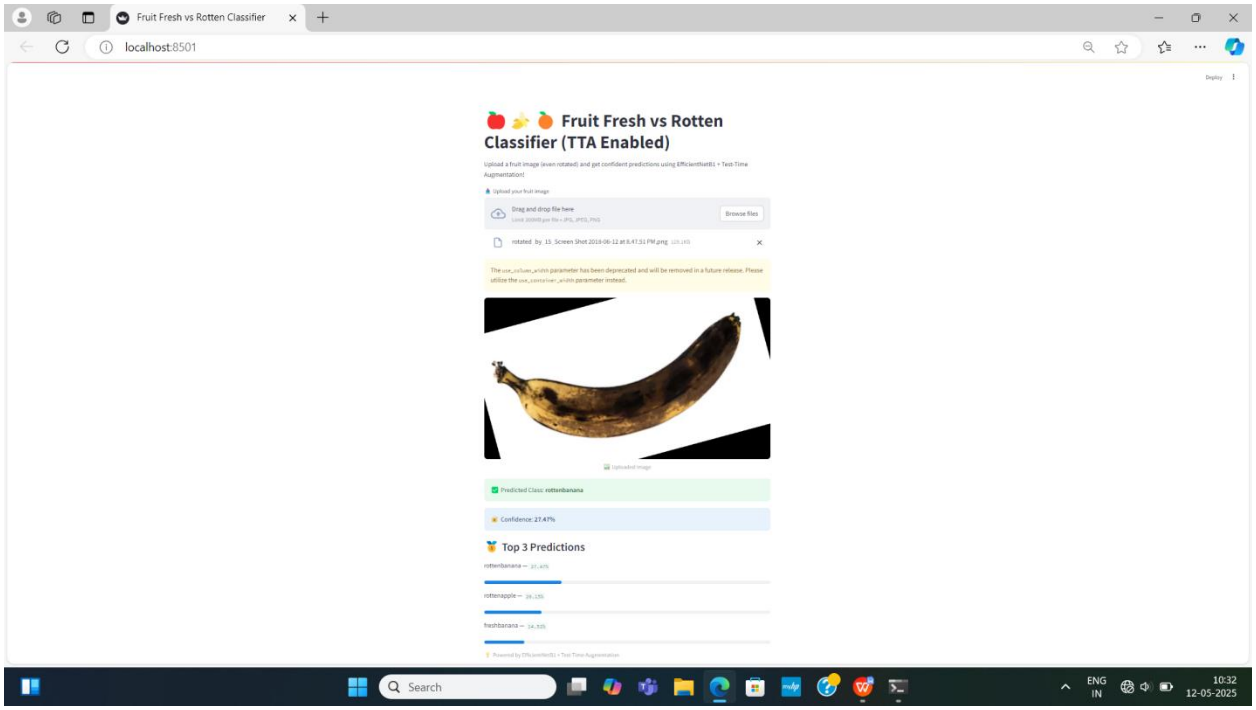
```
            img = cv2.imread(os.path.join(path, img_name))
            img = cv2.resize(img, (100, 100))
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            X_test.append(img)
            Y_test.append(0)  # Label: Fresh (0)

    # Load rotten images (label = 1)
    for rotten_folder in rotten_folders:
        path = os.path.join(test_dir, rotten_folder)
        for img_name in tqdm(os.listdir(path)):
            img = cv2.imread(os.path.join(path, img_name))
            img = cv2.resize(img, (100, 100))
            img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
            X_test.append(img)
            Y_test.append(1)  # Label: Rotten (1)

    return np.array(X_test), np.array(Y_test)

# Load the test dataset
X_test, Y_test = load_test_data()

# Normalize the test images
X_test = X_test / 255.0
```

9.2 SCREEN SHOTS






Fruit Fresh vs Rotten Classifier
localhost:8501
Deploy
Open
input › dataset › test › rottenbanana
Search rottenbanana
Organize
New folder
Home
Gallery
Attachments
Desktop
Documents
Pictures
rotated_by_15_ Screen Shot 2018-06-12 at 8.47.51 PM
rotated_by_15_ Screen Shot 2018-06-12 at 8.48.18 PM
rotated_by_15_ Screen Shot 2018-06-12 at 8.49.59 PM
rotated_by_15_ Screen Shot 2018-06-12 at 8.51.00 PM
rotated_by_15_ Screen Shot 2018-06-12 at 8.51.46 PM
File name:
Custom files
Upload from mobile
Open
Cancel
sh vs Rotten
bled)
predictions using EfficientNetB1 + Test-Time
Browse files
Search
ENG
IN
10:35
12-05-2025


Fruit Fresh vs Rotten Classifier
localhost:8501
Fruit Fresh vs Rotten Classifier (TTA Enabled)
Top 3 Predictions
Search
ENG
IN
10:32
12-05-2025

CHAPTER 10

REFERENCES

1. M. Mukhiddinov, A. Muminov and J. Cho, “Improved classification approach for fruits and vegetables freshness based on deep learning,” Sensors, vol. 22, no. 21, pp. 1–20, 2022

2. N. Saranya, K. Srinivasan and S. K. P. Kumar, “Banana ripeness stage identification: A deep learning approach,” Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 8, pp. 4033–4039, 2022

3. S. S. R. Mamidi, C. A. Munaganuri, T. Gollapall and A. T. V. S. Aditya, “Implementation of machine learning algorithms to identify freshness of fruits,” in Proc. of Third Int. Conf. on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, pp. 1395–1399, 2022

4. N. Aherwadi and U. Mittal, “Fruit quality identification using image processing, machine learning, and deep learning: A review,” Advances and Applications in Mathematical Sciences, vol. 21, no. 5, pp. 2645– 2660, 2022.

5. C. C. Ukwuoma, Q. Zhiguang, M. B. Bin Heyat, L. Ali, Z. Almaspoor et al., “Recent advancements in fruit detection and classification using deep learning techniques,” Mathematical Problems in Engineering, vol. 2022, no. 1, pp. 1–29, 2022.

6. S. Karamchandani, B. Sekhani, K. Nair and K. Shah, “E-nose for shelf-life prediction of climacteric fruits,” in Proc. of 4th Int. Conf. on Computing, Power and Communication Technologies (GUCON), Kuala Lumpur, Malaysia, pp. 1–4, 2021

7. H. Chopra, H. Singh, M. S. Bamrah, F. Mahbubani, A. Verma et al., “Efficient fruit grading system using spectrophotometry and machine learning approaches,” IEEE Sensors Journal, vol. 21, no. 14, pp. 16162– 16169, 2021

8. Z. Fathizadeh, M. Aboonajmi and S. R. Hassan-Beygi, “Classification of apples based on the shelf life using ANN and data fusion,” Food Analytical Methods, vol. 14, no. 4, pp. 706–718, 2021.

9. V. Bhole and A. Kumar, “A transfer learning-based approach to predict the shelf life of fruit,” Inteligencia Artificial, vol. 24, no. 67, pp. 102–120, 2021

10. S. Jana, R. Parekh and B. Sarkar, “Detection of rotten fruits and vegetables using deep learning,”Computer Vision and Machine Learning in Agriculture, vol. 1, no. 1, pp. 31–49, 2021.