AN AUTOMATED PROCESS OF DETECTING FRESH AND ROTTEN VEGETABLES BY USING DEEP LEARNING

BY

Fatin Fuad Chowdhury
ID: 201-15-13618

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering

Supervised By

Faria Nishat Khan
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised By

Md. Sazzadur Ahamed
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

July 2024

APPROVAL

This Project titled **"An Automated process of detecting fresh and rotten vegetables by using deep learning"**, submitted by **Fatin Fuad Chowdhury** to the Department of Computer Science and Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **14th July 2024.**

BOARD OF EXAMINERS

Dr. Md. Fokhray Hossain — **Board Chairman**
Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Amatul Bushra Akhi (ABA) — **Internal Examiner 1**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Mr. Amir Sohel (ARS) — **Internal Examiner 2**
Sr. Lecturer
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

14/7/24
Dr. Shamim H Rippon (DSR) — **External Examiner**
Professor
Department of Computer Science and Engineering
East West University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Faria Nishat Khan, Lecturer, Department of Computer Science and Engineering, Daffodil International University**. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

17.7.24

Faria Nishat Khan
Lecturer
Department of CSE
Daffodil International University

Co-Supervised by:

Md. Sazzadur Ahamed
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Fatin Fuad

Fatin Fuad Chowdhury
ID: 201-15-13618
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to almighty for His divine blessing making it possible for us to complete the final year project/internship successfully.

We are grateful and wish our profound indebtedness to **Faria Nishat Khan**, **Lecturer**, Department of CSE Daffodil International University, Dhaka. Deep Knowledge & keen interest of our supervisor in the field of "*Computer Science and Engineering*" to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to the **Head of the Department of CSE**, **Dr. Sheak Rashed Haider Noori** for his kind help in finishing our project and also to other faculty members and the staff of the Department of CSE, Daffodil International University.

We would like to thank our entire course mate in Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

Classification of fresh and rotten vegetables is essential in our everyday life as it becomes difficult sometimes to identify earlier whether the vegetables are fully fresh or not. There is a possibility of suffering from severe diseases if defect vegetables are consumed. On the other hand, it will also be difficult to import vegetables in foreign countries if freshness is not found which may result in deterioration of economy. For this, we have has been intended to use deep learning method to detect immediately which fresh and rotten vegetables are fresh and which are rotten. Five vegetables are selected which are- Cucumber, Green capsicum, pointed gourd, cabbage and tomato. These are divided into 10 distinct classes according to fresh and rotten categories. Fresh and rotten cucumber, green capsicum, pointed gourd, cabbage and tomato, along with rotten are kept in the tent box for capturing photos which will be good background for photos as only vegetables can be recognized easily. At first, 1096 and 1016 raw image data of fresh and rotten vegetables are captured respectively. This is done by capturing the photos of fresh and rotten vegetables at various angles. After that these were tilted, rotated at the angles different from the angles of photos existed during capturing and resized to increase more in numbers. Among these, 206 images of fresh and 195 images of rotten vegetables are tested and 830 images of fresh and 786 images of rotten vegetables are trained. Here the four models of deep learning InceptionV3, MobileNetV2, VGG16 and Xception model are applied as they can provide good accuracy. InceptionV3 provides the accuracy of 98.50%, accuracy of VGG-16 is 97.01%, MobileNetV2 provides the accuracy of 99.50% and Xception model has the accuracy of 99.75%. Among these four models, Xception is the most accurate model.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
Introduction

1.1 Introduction

The detection of fresh and rotten vegetables is significant for the agricultural and food supply chain, ensuring accurate food quality, safety, and economic development of the country. Here, method of high technology is used to detect the freshness and defect of the collected vegetables. The freshness of vegetables is very important for safety of health as consuming rotten vegetables can cause serious illnesses. Defect may takes place due to keeping in a place of high temperature. It also happens that insect can also cause damage to vegetables [10]. Detecting freshness and defect of vegetables manually takes lot of time and there is also chance of possibility of errors and human fatigue. In order to solve the problem occurred due to manual detection of fresh and rotten vegetables, deep learning method is used so that it can detect fresh and rotten vegetables early in a short period of time correctly and decision can be taken according that. This technology reduces the costs of labor and increases efficiency throughout the food supply chain. We trained deep learning models through large number of images of fresh and rotten vegetables so that we can distinguish those two categories of vegetables fresh and rotten parts in an accurate and errorless way. Ensuring vegetable freshness is crucial in a worldwide market with quality standards. An automated system contributes to maintaining of these standards, and thus improve reputation of exporters and the stability of economy of country. Deep learning models like Convolutional Neural Networks (CNNs) such as InceptionV3, MobileNetV2, VGG-16 and Xception has shown outstanding performance in the image classification of fresh and rotten vegetables. CNN models can collect properties from images automatically and makes them highly suitable to differentiate between which vegetables are fresh and which are rotten. Deep learning method is appropriate starting from grocery, supermarket and upto export markets. The study of deep learning aims to introduce an innovative solution that deals with health concerns, supports economic growth, and modernizes the process of checking vegetable freshness through advanced and upgraded technology.

1.2 Motivation

Deep learning models can handle a huge amount of data by making them suitable for using in various detection activities. Deep learning is useful to recognize fresh and spoiled vegetables which has the possibility of improving food safety, waste reduction, and quality assurance. Accurate identification of vegetable freshness is needed for avoiding health risks, gaining nutrition and minimizing waste. Automated systems due to deep learning can improve accuracy and speed which reduces the possibility of human error and labor expenses. Agricultural and retail sectors face massive economic losses due to occurrence of defect of vegetables during storage and transit. Deep learning models can assist in identifying and separating rotten vegetables and contribute in increasing supply chain efficiency. This ensures high-quality product which ensures customer pleasure and trust in businesses. The use of deep learning in vegetable identification is motivated by the need to improve food safety, minimize waste, optimize supply chain management, and assure high product quality standards for producers, retailers, and consumers.

1.3 Rationale of the Study

The rationale behind the study of "An Automated Process of Detecting Fresh and Rotten Vegetables by Using Deep Learning" lies in analyzing several important challenges and opportunities in the food industry.

Firstly, ensuring the freshness of vegetables using deep learning is much important for health and safety. Defected vegetables causes harm to health when consumed. For this, it is necessary to identify whether vegetables are fresh o rotten so that rotten vegetables can be removed from the supply chain. Traditional methods of checking vegetable freshness causes the wastage of huge time and there can be possibility of error in detection due to tiredness.

Secondly, in the global market, by exporting vegetables successfully maintaining high quality standards is very needed for the. If freshness is not guaranteed, countries and businesses of the countries may face economic losses due to rejection of shipments or loss of reputation. An automated system using deep learning can provide a reliable and consistent method for ensuring the quality of vegetables.

Thirdly, the use of deep learning models offers great technological advantages. These models can process large amounts of data and learn to recognize complex patterns, which makes them highly effective for tasks like helping to understand fresh and rotten vegetables differently. By training these models on diverse datasets, they can achieve high accuracy and can help to adapt to identify different types of vegetables.

Automating detection of fresh and rotten vegetables is a suitable task as it saves money. It removes the need for a lot of manual work, resulting in faster completion of required tasks and reduces mistakes. This makes the supply chain work better and more reliably, which is good for both producers and consumers.

In the end it can be said that, to identify the freshness and defect of vegetables, deep learning is to be used to develop a better system. This new system will be more accurate, efficient, and reliable. It will help ensuring food safety, support the economy by improving exports, and introduce new technology to make the food supply chain better.

1.4 Research Questions

A research question is a clearly focused question that is helpful for study or investigation. It defines about what the researcher wants to find out or understand through their research. A good research question helps to understand the topic properly and gives direction to the study by making it easier to gather relevant information and draw meaningful conclusions. It serves as the foundation for the entire research project, ensuring that the research stays on track and addresses specific aspects of the topic.

- **How much effective are the different deep learning models (InceptionV3, MobileNetV2, VGG16, Xception) in classifying fresh and rotten vegetables?**

- **How does image augmentation (tilting, rotating, resizing) impact the accuracy of deep learning models in detecting fresh and rotten vegetables?**

- **Can the developed deep learning models detect freshness in a variety of vegetable types (cucumber, green capsicum, pointed gourd, cabbage, tomato)?**

- **What economic impacts can be expected from implementing an automated detection system for vegetable freshness in the export market?**

- **How does the real-time performance of the developed system compare to manual identification in terms of speed and accuracy?**

1.5 Expected Output

Vegetable detection using deep learning is more accurate comparing with image recognition. It is effective in both academic research and industry. When the image datasets are properly trained, the four models can be used to find out the difference between fresh and rotten vegetables effectively. Among the four models which were tested, Xception model was found to be the most accurate which shows the accuracy of 99.75%.

CHAPTER 2

Background

2.1 Preliminaries/Terminologies

Deep learning is a type of artificial intelligence that uses neural networks with many layers so that it user can learn from large amounts of data. These networks can recognize patterns and make decisions, much like the human brain. Then the concept of a Convolutional Neural Network (CNN) is known. A CNN is a specific kind of deep learning model that are used for processing images. It works by scanning an image with filters so that the features such as edges, colors, and textures can be detected which helps in identifying objects within the images. From this the correct information will be found. Image Classification is the process of training a model in order to recognize and categorize objects in images. In this study, the goal is to classify vegetables as either fresh or rotten. Another term is Dataset, which refers to the collection of images used to train and test the model. A good dataset includes a wide variety of images to help the model learn effectively. Finally, the term Accuracy is mentioned, which measures how well the model can correctly classify the images into separate categories. High accuracy means the model is making correct predictions most of the time.

2.2 Related Works

Roul et al. [1] showed a new deep learning model can detect the quality of Vegetables and vegetables in a shot with 99% accuracy. It performs best in bright light against a plain background. The mobile-based methodology considers color, texture, and flaws when determining costs. An app will allow farmers and sellers to input data to help improve the system and establish a national quality register. Tasmina [2] et al. stated that identifying fresh vegetables is difficult since chemical use affects their quality. A novel system based on DenseNet201 can categorize vegetables as fresh, old, or rotten with 98.56% accuracy from photographs. The current dataset only comprises photos of the exterior shell and does not take into account temperature or surroundings. Future enhancements will include these aspects as well as internal photos, and the system will be accessible via mobile and web apps. Yuan [3] et al. presents a novel method for detecting the freshness of vegetables and Vegetables using deep learning. It reduces features from three pre-trained models (GoogleLeNet, DenseNet-201, and ResNeXt-101) using PCA. The approach obtained

96.98% accuracy in tests. While promising, additional research is required to make it feasible for general use. Rehanul [4] et al. introduces FreshDNN, a bespoke CNN model developed to detect the freshness of Vegetables and vegetables, which is critical for reducing deterioration and preventing foodborne infections. FreshDNN was trained and tested against six pre-trained CNN models using a dataset compiled from Kaggle and Mendeley. FreshDNN outperformed the other models in terms of accuracy, computation speed, and model size. Future development includes merging Explainable AI and vision transformer technologies to improve performance even further. Kazi [5] et al. stated that detecting rotten fruits and vegetables is very important in the agricultural industry. Computer vision is useful for automating the detection of damaged and fresh produce. Recently, farmers have found that using computer vision and image processing helps a lot, especially in quality control by identifying rotten produce. Farmers struggle to distinguish fresh from rotten produce because this task is usually done by people, who can get tired. Robots, however, do not get tired. This study proposes a method to reduce human effort and work time by identifying defects in agricultural products. Rotten vegetables and fruits can affect healthy ones if not identified in time. Therefore, the suggested method aims to prevent the spread of rot. The model detects fresh and decaying produce based on photos of fruits and vegetables. The study tested six types of vegetables and six types of fruits. It discusses different image processing methods for categorizing rottenness, using CNN, KNN, and SVM to analyze the photos. The suggested model's efficiency was tested on Google and Kaggle datasets, with the CNN model showing the highest accuracy of 95%. Jana et al. [6] used CNN to detect rotten or damaged fruits and vegetables. Modern people care a lot about food safety, especially because rotten or damaged fruits and vegetables can harm health and lower food quality in processing industries. These bad fruits and vegetables can also spoil the fresh ones they touch in inventories and supermarkets. Therefore, it's crucial to detect and remove them early. Using automated systems with image analysis and deep learning can be very effective for this purpose. The CNN model is trained to classify fresh from non-fresh produce and has shown high accuracy and F1 scores when tested on two different datasets. Accuracy of first dataset is 98.53% and that of second is 98.46%. The F-1 scores are 99% and 99.19%. Zhou et al. [7] developed a robust method for segmenting and grading broccoli heads in field conditions using the Improved ResNet deep learning algorithm. It outperformed other approaches (GoogleNet, VggNet, and ResNet) in segmentation and grading accuracy. The model was effective under various light intensities and noise conditions, demonstrating its robustness. Accurate segmentation allows for the extraction of traits like Above Ground Biomass (AGB), yield, and biological rhythm, making this algorithm a powerful tool for large-scale, non-invasive phenotyping analysis and potentially aiding future broccoli breeding research. Yaodi [9] et al. aimed to automate the detection of surface defects in fresh-cut cauliflower using a convolutional neural network (CNN) and transfer learning. They optimized the model by adjusting training hyperparameters and freezing different numbers of layers, integrating parameters from ImageNet with their experimental training. The improved MobileNet model achieved a high test accuracy of 98.85% and F-1 score of 98.87% with optimal hyperparameters (dropout = 0.5, learning rate = 0.001) and 80 frozen layers. Comparisons with other CNN architectures (VGG19, InceptionV3, NASNetMobile) showed MobileNet's superiority. Future work includes expanding the dataset, refining the model for mobile deployment, and enhancing its industrial applicability in fresh-cut cauliflower quality inspection. Mahmud et al. [10] reviews the use of machine vision (MV) and image processing (IP) in agriculture to detect defects, assess quality, and classify Vegetables and vegetables. It highlights the benefits of computer vision systems (CVS) in providing accurate, rapid, and non-destructive evaluations compared to traditional methods. The study emphasizes the need for advanced technologies like deep learning and hyperspectral imaging to improve detection accuracy and speed. It calls for further research to develop better algorithms, faster processing techniques, and comprehensive image databases to enhance automatic grading applications in the agricultural sector.

2.3 Comparative Analysis and Summary

In "An Automated Process of Detecting Fresh and Rotten Vegetables by Using Deep Learning," it is known how different deep learning models can effectively distinguish between fresh and rotten vegetables. This research is crucial because it addresses the challenge of identifying vegetable freshness accurately, which is important for both consumer health and economic reasons, especially in the export market. It is not possible to manually detect fresh and rotten vegetables for a long time as it is difficult to complete monotonous tasks [6].

We collected a large dataset of images containing fresh and rotten vegetables such as cucumbers, green capsicums, pointed gourds, cabbages, and tomatoes. These images were crucial for training the deep learning models. They employed four different models: InceptionV3, MobileNetV2, VGG16, and Xception, all known for their capabilities in image recognition tasks.

After training and testing these models, we found notable differences in their performance. InceptionV3 achieved an accuracy of 98.50%, while VGG16 had an accuracy of 97.01%. MobileNetV2 performed even better with an accuracy of 99.50%, and Xception emerged as the most accurate model with an impressive accuracy of 99.75%. This means that Xception was the most reliable in correctly classifying whether vegetables were fresh or rotten. The study also explored the impact of image augmentation techniques such as tilting, rotating, and resizing. These techniques were used to enhance the dataset by creating variations of the original images. This augmentation proved beneficial as it helped the models generalize better to different angles and conditions of vegetable photos, ultimately improving their accuracy. The research demonstrated that deep learning models, especially Xception, can be highly effective in automating the detection of vegetable freshness. This automated process not only ensures better quality control but also has significant implications for health safety and economic efficiency, particularly in industries reliant on the export of fresh produce. The findings suggest that integrating such technology could lead to more reliable and efficient vegetable quality assessment systems in the future.

TABLE 2.3.1: Comparative Analysis Table

Study	Image type	Models used	Accuracy
Roul et al. (2020)	Qualitative vegetable image	Deep Learning Model	99% Accuracy
Tasmina et al. (2022)	Fresh and rotten vegetable imags	Deep Learning Model DenseNet201	98.56% Accuracy
Yuan et al. (2024)	Fresh vegetable images	Deep Learning Model	96.98% Accuracy

Rehanul et al. (2023)	Fresh vegetable images	Deep Neural Network FreshDNN	Validation Accuracy of FreshDNN is 97.8% and Test Accuracy is 97.64%.
Kazi et al. (2023)	Fresh and rotten fruits and vegetable detection	Deep Learning KNN, SVM	95% Accuracy
Jana et al. (2021)	Defect fruits and vegetables	Convolutional Neural Network (CNN)	Accuracy of 1st dataset is 98.53%. Accuracy of 2nd dataset is 98.46%.
Zhou et al. (2020)	Images of broccoli head	ResNet deep learning algorithm	89.6% Accuracy
Yaodi et al. (2022)	Images of a portion of fresh cauliflower	Convolutional Neural Network (CNN) such as-VGG19, InceptionV3, NASNetMobile, MobileNet and transfer learning	99.27% Accuracy
Fatin et al.	Fresh and rotten vegetables	CNN models such as InceptionV3, MobileNetV2, VGG-16 and Xception	Accuracy of InceptionV3=98.50%, Accuracy of MobileNetV2=99.50%, Accuracy of VGG-16=97.01%, Accuracy of Xception=99.75%

Among the accuracies of the models performed by other authors and us, ours is the highest.

2.4 Scope of the Problem

The scope of the problem is the difficulty of accurately determining the freshness of vegetables using advanced technology.

Firstly, the need for such a system comes from the difficulty in visually differentiating between fresh and rotten vegetables, which is hard to detect with the naked eye. This is crucial because consuming rotten and defected vegetables can lead to health issues, and so it is essential to ensure freshness of vegetables before consumption.

Secondly, in industries where vegetables are exported, ensuring freshness of vegetables is vital for maintaining consumer trust and following international quality rules. The ability to automate this process using deep learning models offers a promising solution to improve accuracy and efficiency compared to traditional manual inspection methods.

The scope also extends up to the technical aspects of implementing deep learning models. This includes gathering large datasets of images containing both fresh and rotten vegetables, training the models to recognize patterns that distinguish between the two states, and optimizing their performance to achieve high accuracy.

Furthermore, the study explores the potential economic impacts of implementing such automated systems. By reducing the risk of exporting spoiled vegetables, countries and businesses can enhance their competitiveness in global markets and minimize economic losses due to rejected shipments or lower consumer confidence.

Overall, the scope of this problem encompasses technological innovation, health considerations, quality assurance in food production, and economic implications. Addressing these aspects through automated detection systems not only improves food safety and consumer health but also strengthens the efficiency and reliability of vegetable supply chains globally.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Research Methodology:

Research methodology of project is the most important aspect of its work. It is a systematic framework for conducting research, which includes the strategies, techniques, and tools used to collect, analyze, and interpret data. It is like a detective's toolbox. It stores all the tools and steps a curious mind uses to find out the truth about a topic. The detective which is researcher first figures out what they want to find the research question. Then, they choose the data collection methods like interviews, surveys, or experiments to gather clues which is data. Finally, they carefully examine the data analysis to find out conclusion about the case which is research findings. A good research methodology ensures the detective uses the right tools and examines the clues fairly, leading to a reliable solution to the mystery. A large collection of vegetable images is gathered to create a dataset. This dataset includes images of both fresh and rotten vegetables. Then the images are preprocessed to make them suitable for training a deep learning model. This might include resizing the images, normalizing them, and augmenting the dataset to improve the model's performance. A convolutional neural network (CNN) is then designed and trained using this dataset. The CNN learns to recognize features that distinguish fresh vegetables from rotten ones. After training, the model is tested on a separate set of images to evaluate its accuracy. This helps to ensure that the model can reliably classify new images of vegetables as fresh or rotten. Finally, the results are analyzed, and the model's performance is assessed. Any necessary adjustments are made to improve accuracy, and the methodology may be refined for future research. It includes defining the research problem, establishing objectives, and developing hypotheses. A research project advances step by step so that the target goals can be reached. With the right approach and preparation, it is possible to learn new things along the way. In our study, we used various methods to work with the vegetable recognition dataset. These methods included analyzing the data, preparing it for use, and enhancing it to improve the results. Here the Trained Models are the VGG16, InceptionV3, Xception and MobileNetV2.

3.1.1 Workflow for Vegetable detection System:

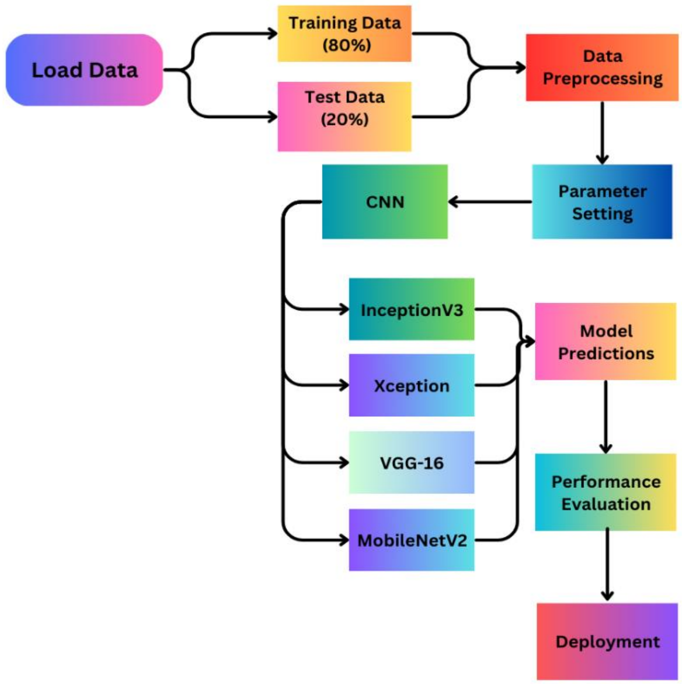

Fig.3.1.1.1 Proposed Workflow

This figure shows the process of developing and deploying deep learning model for image recognition. The process starts by loading the data, which is then split into training data (80%) and test data (20%). The data undergoes preprocessing to make it suitable for the model. Next, the parameters for the model are set.

The main part of the process involves utilizing Convolutional Neural Networks (CNNs). There are different types of CNN architectures mentioned, including InceptionV3, Xception, VGG16, and MobileNetV2. These models are used to make predictions based on the data.

The predictions are then evaluated to determine the model's performance. If the model meets the desired criteria, it is deployed for practical use. The entire process ensures that the model is trained, tested, and fine-tuned before being deployed.

3.1.2 Data Collection:

Data collection is the systematic process of gathering information from various sources to answer specific research questions or objectives. Data collection is the process of gathering information or observations for a specific purpose. In research, this often involves collecting relevant data that will be analyzed to answer questions or test hypotheses. It can include gathering data from sources like surveys, experiments, or existing databases. The goal is to gather enough data that is accurate and representative to draw meaningful conclusions or insights. It is crucial because the quality and quantity of the data gathered directly impact the reliability and validity of the research findings. It is needed for any analysis. Accuracy would be perfect if data sets are more genuine. Effective data collection is critical for gathering accurate and comprehensive information, which serves as the foundation for analysis and informed decision-making in research.

Fig 3.1.2.1 Sample of data

We extracted both fresh and rotten vegetables for our project as shown in **Fig 3.1.2.1**. Cucumber, tomato, cabbage, pointed gourd, and green capsicum are the five varieties of new and rotting vegetables we used. We bought them from supermarket. We used a tent box to capture photographs of vegetables for this project. Then by processing, the fresh vegetables are made rotten and the photos of rotten are captured in different angles. After that, the images are resized, tilted at different direction, rotated at various angles.

3.1.3 Data Preprocessing:

Data preprocessing is the initial step in analyzing data where raw data is transformed into a more understandable and usable format. This process involves cleaning the data by removing or correcting any errors or inconsistencies. It also includes transforming the data into a standardized format, such as scaling numerical values or converting categorical variables into numerical ones. It aims to prepare the data so that it can be effectively analyzed by deep learning algorithms or statistical methods. By doing this, researchers can ensure that the data is accurate, complete, and ready for further analysis to extract meaningful insights. It is the method of manipulating and converting raw data into an accessible format so that data can be more perfected. It is an important step in preparing raw data for analysis because it cleans and transforms it into usable formats. This procedure entails several steps, including handling missing values, correcting errors, normalizing or scaling data, and encoding categorical variables. A successful preprocessing step improves the quality of the data and increases the precision and effectiveness of any further analysis or deep learning models. Files are rescaled and RGB is made up of training samples.

3.1.4 Data Augmentation:

Data augmentation is a technique used in machine learning and deep learning to artificially increase the size and diversity of a dataset. It is used to create image into new ones. The goal of data augmentation is to improve the generalization and robustness of deep learning models. By exposing the model to a wider range of variations in the data during training, it learns to better handle different scenarios and variations it may encounter in real-world applications. Using different types of transformations, including rotations, flips, scaling, cropping, and noise addition, to the original data, new data points are created throughout this procedure. By offering more diverse training examples, data augmentation enhances the generalization and resilience of deep learning models in image processing. It improves the model's performance on unknown data and helps in avoiding overfitting by growing the sample artificially. Datasets are in the paper to obtain adequate precision. By doing so, new variations of the original data are created without collecting additional real-world data. This helps in reducing overfitting and improving the model's performance when applied to new, unseen data. Overall, data augmentation is a powerful tool used to enhance the effectiveness and reliability of deep learning models. After data augmentation, 401 images and 1016 images are found by data augmentation.

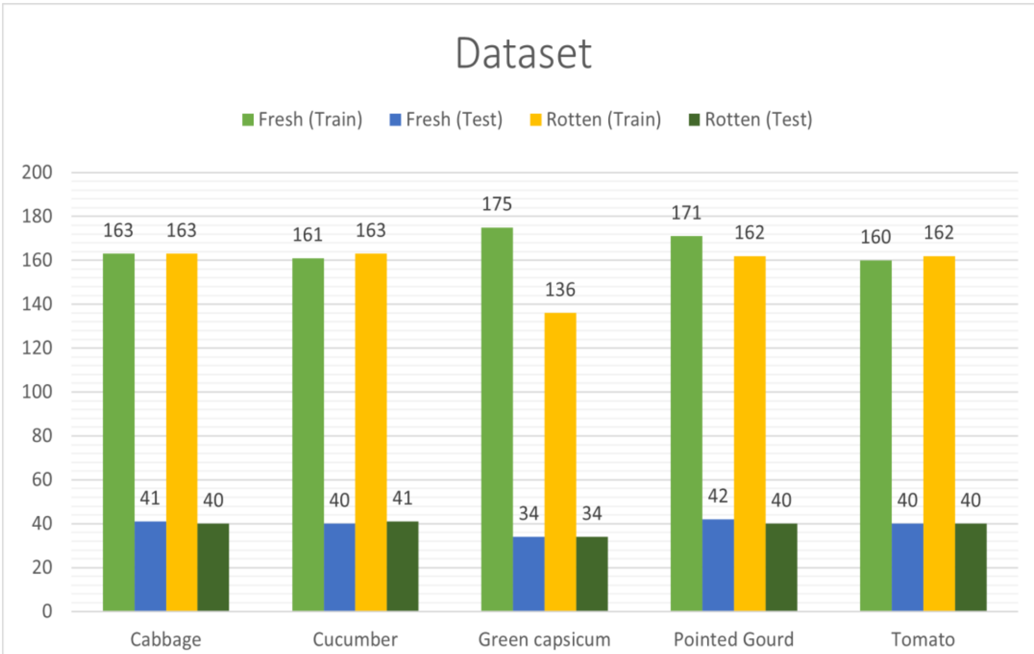

Fig. 3.1.4.1 Train & Test data Bar chart.

It shows the datasets which are training and test number bar charts, as shown in Fig. 3.1.3.4. Four types of vegetables are obtained in the above Bar chart. These are - green represents fresh vegetables training data, blue represents fresh test data, yellow represents rotten ride, and dark green represents test image datasets of rotten vegetables.

3.1.5 Training the model using the training dataset :

Four different convolutional neural network models are trained on our training dataset which includes 1616 pictures of vegetables. I've chosen the following models:

3.1.5.1 INCEPTIONV3:

InceptionV3 is a deep learning model designed for image recognition tasks. It is known for their high performance and efficiency. The InceptionV3 model uses a unique architecture that includes multiple types of layers working together. These layers can process different parts of an image at the same time, allowing the model to capture a wide range of features. For example, it can detect small details like edges and textures, as well as larger patterns like shapes and objects. One of the key features of InceptionV3 is its use of "Inception modules." These modules are like small sub-networks within the larger network, each analyzing different aspects of the image. This approach helps the model to be both deep and wide, making it very effective at understanding complex images. InceptionV3 is also designed to be efficient, meaning it can achieve high accuracy without requiring excessive computational resources. This makes it suitable for use in real-world applications where both speed and accuracy are important. InceptionV3 is a powerful and versatile deep learning model that excels at recognizing and classifying images. It has been widely used in various fields, including medical image analysis, object detection, and more.

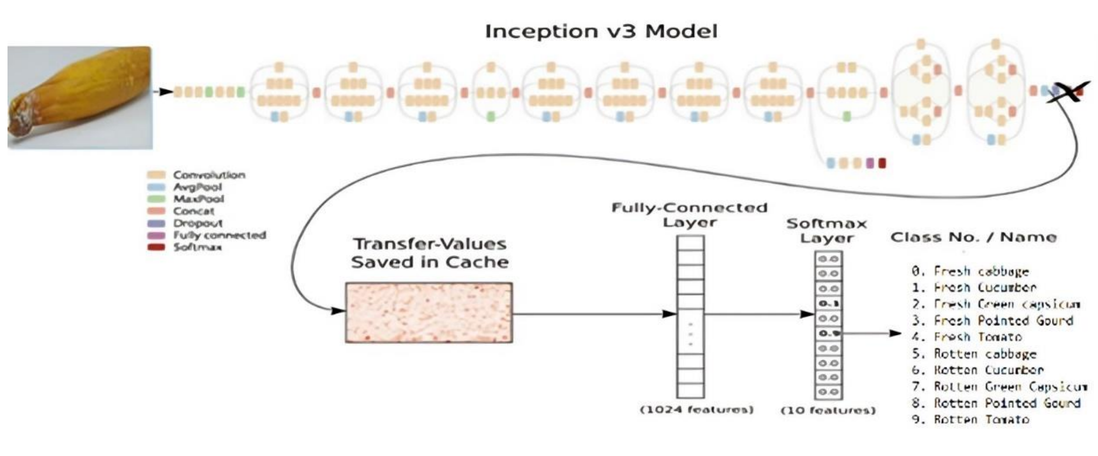

Fig.3.1.5.1.1 InceptionV3 architecture.

3.1.5.2 XCEPTION:

The Xception model is a deep learning neural network used for image recognition tasks. Its name stands for "Extreme Inception," as it is an improved version of the Inception model. Xception uses a unique architecture called "depthwise separable convolutions." This means it processes the spatial details (like shapes and edges) and the depth details (like colors and textures) of an image separately. This separation makes the model more efficient and faster, as it reduces the number of calculations needed. The model starts by breaking down the image into small parts to recognize basic features. Then, it combines these features to understand more complex patterns and objects in the image. This step-by-step process helps Xception achieve high accuracy in identifying and classifying images. Xception is known for being both powerful and efficient, making it a popular choice for many image recognition applications, such as detecting objects in photos, diagnosing medical images, and more. Its innovative design helps it perform well while using less computational power compared to other complex models.

3.1.5.3 VGG-16:

The VGG-16 model is a deep learning neural network used for image recognition tasks. Its name stands for Visual Geometry Group, which is the research group at the University of Oxford that developed it. The "16" in VGG-16 refers to the fact that it has 16 layers in its architecture. VGG-16 is designed to analyze images by breaking them down into smaller parts and gradually building up a detailed understanding. It uses a series of convolutional layers, which are layers that apply filters to the image to detect features like edges, textures, and shapes. These convolutional layers are followed by pooling layers, which reduce the size of the data while retaining important information, making the model more efficient. One of the key features of VGG-16 is its simplicity and uniformity. It uses very small filters (3x3) consistently throughout the network, which helps to capture fine details in the images. Despite its simplicity, VGG-16 is very powerful and has been shown to achieve high accuracy in image classification tasks. VGG-16 is widely used because it balances good performance with a straightforward design, making it easier to understand and implement compared to more complex models. It has been applied in various fields, such as object detection, facial recognition, and medical image analysis.

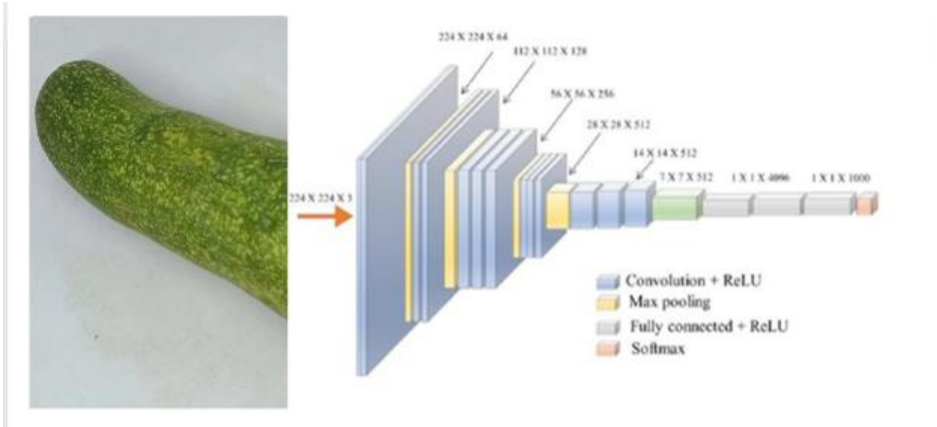

Figure 3.1.5.3.1: VGG-16 architecture.

There are three convolutional layers that are totally connected, the first two with 4096 channels each and the third with seven channels, one for each class. The soft peak layer is the uppermost layer. Many of the network's totally linked levels are configured in the same way.

3.1.5.4 MOBILENETV2:

MobileNetV2 is a deep learning model designed for image recognition tasks, especially optimized for mobile and resource-constrained devices. It is known for being both efficient and lightweight, meaning it can run quickly and use less computational power while still achieving good accuracy. The model uses a technique called "depthwise separable convolutions," which splits the process of filtering and combining image data into two simpler steps. This reduces the amount of computation needed, making the model faster and more efficient. One of the key features of MobileNetV2 is the "inverted residuals" with linear bottlenecks. This design helps to preserve important information while reducing the amount of data the model needs to process, further enhancing efficiency. It is particularly useful for applications on mobile devices, like smartphones and tablets, where processing power and memory are limited. It can be used for various tasks such as object detection, image classification, and even real-time video analysis. It is TensorFlow's first mobile computer vision model. The MobileNetV2 model uses depth-wise separable convolutions, a sort of factorized convolution that converts a pointwise convolution into a depth wise convolution and a conventional convolution into a depth wise convolution (11 convolutions). MobileNetV2's depth-wise convolution uses a single filter for each input channel. In terms of values, the results are blended using an 11-convolution. Convolution in depth is known as development. A traditional convolution filters and integrates inputs in a single step to generate a new set of outputs. The depth-wise separable convolution splits this into two layers, one for filtering and one for mixing. Factorization limits the number of variables. MobileNetV2 uses depth-wise separable convolutions. Compared to a network of regular convolutions of the same depth, it significantly reduces the number of parameters. As a result, lightweight deep neural networks are created.

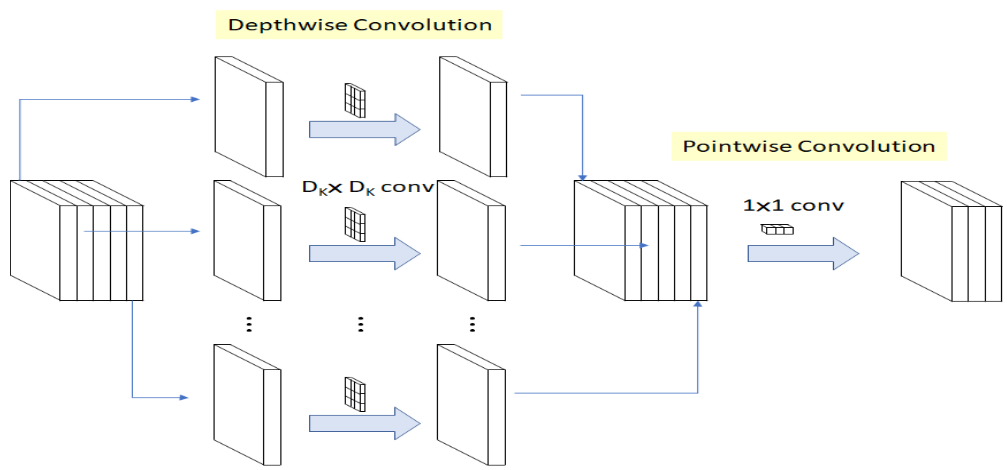

Figure 3.1.5.4.1: MobileNetV2 Identity Mapping.

3.1.6 Evaluation of the model:

Its goal is to calculate generalization accuracy of a model which is for the future data. It is the process of evaluating a trained deep learning performance of model on a different dataset which is referred to as the test set, in order to ascertain how well it generalizes to new and unknown data. Evaluating a model means checking how well it works in making predictions. This involves several steps to ensure the model is accurate and reliable. First, we split our data into two parts: training data and test data. The model learns from the training data, and then we test it using the test data to see how well it performs on new, unseen examples. Next, we use different metrics to measure the model's performance. A classification model is evaluated using three key criteria which are accuracy, precision, and recall. The fraction of correct forecasts for the test data is known as accuracy. To determine the answer, the number of correct predictions is divided by the total number of predictions. Accuracy tells us the percentage of correct predictions, and other metrics like precision, recall, and F1 score, which give us more detailed insights into the model's strengths and weaknesses. We also create a confusion matrix, which shows how often the model makes correct and incorrect predictions for each category. This helps identify any specific areas where the model might be struggling. Finally, we might use techniques like cross-validation, where we repeatedly split the data into training and test sets in different ways, to get a more robust evaluation of the model's performance. Evaluating a model helps us understand how well it works, where it needs improvement, and whether it is ready to be used for real-world tasks.

At first we tried with different models of deep learning but among them, InceptionV3, MobileNetV2, VGG-16 and Xception model gave the accurate results. The model is evaluated using the Confusion matrix, Accuracy, F1-score, Precision, Recall, and plot diagram.

The model architecture of our performed 4 models is shown below in tabular form:

Table – 3.1.6.1: Explanation of Model Architecture

Feature	**Model 1: InceptionV3**	**Model 2: MobileNetV 2**	**Model 3: VGG16**	**Model 4: Xception**
Base Model	InceptionV3	MobileNetV 2	VGG16	Xception
Weights	imagenet	imagenet	imagenet	imagenet
Include Top	FALSE	FALSE	FALSE	FALSE
Input Shape	Specified by input_shape variable	Specified by input_shape variable	Specified by input_shape variable	Specified by input_shape variable
Freezing Base Layers	Yes	Yes	Yes	Yes
Flatten Layer	Yes	Yes	Yes	Yes
Batch Normalization	Yes	Yes	Yes	Yes
Dense Layer 1	512 units, ReLU	512 units, ReLU	512 units, ReLU	512 units, ReLU
Dropout 1	0.5	0.5	0.5	0.5
Dense Layer 2	256 units, ReLU	256 units, ReLU	256 units, ReLU	256 units, ReLU
Dropout 2	0.5	0.5	0.5	0.5
Output Layer	output_classes units, Softmax activation	output_class es units, Softmax activation	output_classes units, Softmax activation	output_classes units, Softmax activation
Optimizer	Adam with Exponential Decay learning rate	Adam with Exponential Decay learning rate	Adam with ExponentialDecay learning rate	Adam with ExponentialDecay learning rate
Initial Learning Rate	0.001	0.001	0.001	0.001
Decay Steps	10000	10000	10000	10000
Decay Rate	0.9	0.9	0.9	0.9

The above table of Explanation of Model Architecture compares four deep learning models, each based on a different pre-trained architecture, namely InceptionV3, MobileNetV2, VGG16, and Xception. These architectures serve as the foundation for building and fine-tuning models for specific tasks, likely involving classification given the context provided.

For each model, the pre-trained weights are taken from the ImageNet dataset, a common practice to leverage existing knowledge and improve performance on new tasks. The top layers, which originally performed classification in the pre-trained models, are excluded. Instead, custom layers tailored to the new task are added.

The input shape for the models is specified through a variable called input_shape. This allows flexibility in determining the input dimensions according to the specific requirements of the task. The base layers of each model are frozen, meaning their pre-trained weights remain unchanged during the training process. This approach helps in preserving the valuable features these models have already learned.

Each model includes a flatten layer, which converts the multidimensional data into a one-dimensional vector, facilitating further processing by dense layers. Additionally, batch normalization is applied, which helps in stabilizing and accelerating the training process by normalizing the inputs of each layer.

The custom part of the models includes two dense layers. The first dense layer consists of 512 units with a ReLU activation function, which introduces non-linearity into the model. Following this layer, a dropout rate of 0.5 is applied to prevent overfitting by randomly deactivating 50% of the neurons during training. The second dense layer comprises 256 units, also with ReLU activation, and is followed by another dropout layer with the same rate.

The output layer for each model has a number of units equal to the number of output classes, using a Softmax activation function. This configuration is typical for classification tasks, where the goal is to assign each input to one of several classes.

For optimization, all models utilize the Adam optimizer, which is known for its efficiency and effectiveness in training deep learning models. The learning rate starts at 0.001 and decays exponentially. Specifically, every 10,000 steps, the learning rate is reduced by multiplying it by 0.9, ensuring the training process converges smoothly over time.

In summary, this table outlines a comprehensive setup for fine-tuning pre-trained models on new tasks. It incorporates techniques like freezing base layers, adding custom dense layers, applying dropout for regularization, and using an adaptive learning rate to enhance the training process and improve model performance.

3.1.7 PARAMETER SETTING

Setting parameters in a model means choosing specific values for the settings that monitor how the model learns and makes predictions. These parameters are crucial because they can significantly affect the model's performance. At first, we have hyperparameters, which are settings chosen before training the model. Examples include the learning rate, which determines how quickly the model updates its knowledge, and the number of layers or neurons in a neural network. Choosing the right hyperparameters often involves some trial and error, as well as techniques like grid search or random search to find the best values. During the training process, the model also learns parameters from the data. These are values like weights in a neural network, which are adjusted as the model learns to make better predictions. The training process involves optimizing these parameters to minimize the difference between the model's predictions and the actual results. Setting parameters correctly is essential for building a model that learns effectively and performs well on new data. It requires a combination of initial choices and adjustments based on how the model performs during training.

We conducted our investigation using a dataset of fresh and rotten Vegetables. We divided our dataset into two sections: 80% which is for training and 20% for testing. At the same time, preparation and validation takes place. We evaluated the influence of parameters during training and tweaked them to create an accurate model.

The models were trained on a variety of fresh and decaying vegetables, and their precision was determined on a test set.

The parameters for the suggested CNN model are as follows:

TABLE – 3.1.7.1: Training Parameters

Batch-Size	32
Epoch	10
Training	80%
Testing	20%
Output Class	10
Input Shape	224x224x3
Train Samples	1616
Test Samples	401

3.1.8 Effect of Batch Size

Batch size in deep learning refers to the number of training examples the model processes before updating its parameters. Choosing the right batch size can significantly impact the model's training and performance. When using a small batch size, the model updates its parameters more frequently. This can lead to more accurate and fine-tuned updates, potentially helping the model learn better. However, it also means the training process can be slower and more computationally intensive. With a large batch size, the model processes more examples at once, which can make the training faster and more efficient. However, the updates to the parameters of models are less frequent and may be less precise, potentially leading to less effective learning. The right batch size strikes a balance between training speed and learning effectiveness. It is often chosen based on the available computational resources and the specific requirements of the task. Experimenting with different batch sizes helps to find the most suitable one for a given model and dataset. The batch size determines the number of input samples given to the network. Another component that influences classification accuracy is batch size. The higher the batch size, the longer it takes to train the dataset, resulting in lower model precision and memory requirements. As a result, when choosing on batch size, we must exercise prudence. We perform our simulations in batches of 32.

3.1.9 Effect of Number of epochs

Epochs are defined simply as the number of iterations. The number of epochs is a hyperparameter that determines how many times the learning algorithm traverses the entire training dataset. The internal model parameters could be updated once every epoch for each sample in the training data. The number of epochs in deep learning refers to the number of times the model goes through the entire training dataset. Choosing the right number of epochs is important for the model's performance. If we use too few epochs, the model might not learn enough from the data, resulting in underfitting. This means it won't perform well on new, unseen data because it hasn't had enough time to capture the patterns in the training data. If we use too many epochs, the model might learn too much from the training data, resulting in overfitting. This means it will perform very well on the training data but poorly on new data because it has memorized the training examples rather than learning general patterns. The goal is to find a balance where the model has learned enough to generalize well to new data without overfitting. This often involves monitoring the model's performance on a validation set and stopping the training when the performance stops improving or starts to decline.

3.1.10 Effect of optimizers

Optimizers in deep learning are algorithms that adjust the model's parameters to minimize the error between the model's predictions and the actual results. Choosing the right optimizer can greatly affect how well and how quickly a model learns. Different optimizers have different strategies for updating the model's parameters. For example, Stochastic Gradient Descent (SGD) updates parameters using a fixed learning rate and can be effective but may be slow and get stuck in local minima. This helps the model learn faster and can avoid some issues like getting stuck in local minima. The choice of optimizer affects the speed of learning, the stability of the training process, and the model's final performance. Finding the right optimizer often involves experimenting with different ones to see which works best for a specific task and dataset. Optimizers improve the efficiency of our model by modifying weight values to minimize the loss function. Our goal is to reduce the loss of our neural network by optimizing its parameters. The loss function computes loss by utilizing a neural network to match the real and predicted values. We applied three optimizers: Adam, SGD, and RMSprop. Adam is a more advanced optimizer that adapts the learning rate for each parameter based on how it changes over time.

3.1.11 Confusion Matrix:

A confusion matrix is a table that depicts how well a classification model performs given known true values on a set of test data. It shows the number of correct and incorrect predictions which are made by the model and organized by actual vs predicted classes. The matrix includes four key elements: True positives (correctly predicted positive cases), True negatives (correctly predicted negative cases), False positives (incorrectly predicted positive cases), and False negatives (incorrectly predicted negative cases). By analyzing these elements, the confusion matrix provides insights into the model's accuracy, precision, recall, and overall performance, helping to identify areas for improvement.

We utilized the following indicators to determine our infrastructure:

(for all classes) = (Number of Correctly Classified Images)/(Number of All Images).

Accuracy (for each class) = (TP+TN)/(TP+FP+TN+FN)…………….(i)

Specificity = TP/(TN + FP).

Recall = TP/(TP + FN) …………….(ii)

Precision=TP/(TP + FP) …………….(iii)

In the above equations, TP denotes the number of correctly categorized images in a class (true positive) [23]. The terms FP (false positive) and FN (false negative) denotes the number of incorrectly classified photos and misidentified images from different classes.

False negatives are those that do not correlate to a class, while true negatives are not classified.

3.1.12 Accuracy:

Classification model accuracy is the model's predicted percentage. Accuracy is the percentage of properly classified pictures to total samples. Accuracy in deep learning is a metric used to measure how well a model's predictions match the actual results. It is calculated as the ratio of the number of correct predictions to the total number of predictions made. It is the way to understand a model's performance, especially when dealing with balanced datasets where each class has a similar number of examples. For imbalanced datasets in which some classes are much more common than others, accuracy alone might not give a complete picture of the model's performance, and other metrics like precision, recall, or F1 score might be needed.

If a model has high accuracy, it means it makes correct predictions most of the time. The equation of model accuracy is -

Accuracy=(TP+TN)/(TP+TN+FP+FN)………..(i)

3.1.13 Precision:

Precision in deep learning is a measure of how accurate the positive predictions of a model are. It tells us the proportion of positive predictions made by the model that are actually correct. It is the ratio of true positive predictions to the total number of positive predictions made by the model. Precision is the chance of a positive label and how many are positive. Precision shows how many accurately anticipated instances were positive. Precision is helpful when FP mattersmore than FN.

Mathematically, the equation of Precision is:

Precision=TP/(TP+FP)………..(iii)

3.1.14 Recall:

Recall or Sensitivity measures how many positive expected occurrences were classified accurately. Recall in deep learning is a measure of a model's ability to identify all the relevant instances in a dataset. It tells us how well the model captures all the true positives. It is helpful when FN prevails over FP. F1-score is a further metric for classification accuracy that takes into account both recall and precision. As precision and recall are harmonic means so, the F1 score provides an elaborative understanding of these two metrics. It reaches its optimum when Precision and Recall areequal.

To figure it out, use the equation below:

Recall=TP/(TP+FN) ………..(ii)

3.1.15 F-1 Score:

F1 score is a recall-precision metric of categorization accuracy. As F1-score is the harmonic mean of Precision and Recall it gives more weight to the lower of the two values. It gives a complete picture. This means that both precision and recall need to be high for the F1 Score to be high. Precision and Recall are optimal when equal.

Equation of F-1 Score is given below:

F-1 Score=2*(Precision*Recall)/(Precision+Recall) ………..(iv)

High F1 Score indicates that the model performs well in making correct positive predictions and in identifying all relevant positive cases. This is useful in situations where there is an uneven distribution of classes, and it is also important to consider both false positives and false negatives. The F1 Score provides a balanced measure of a model's accuracy which helps to ensure that neither precision nor recall is disproportionately high at the expense of the other.

3.1.16 Validation Loss:

Validation loss is a metric used in machine learning and deep learning to assess a model's performance on a separate validation dataset while training. It aids in determining how well the model generalizes to new, previously unseen data.

3.1.17 Training Loss:

Training loss is a measure of how well a machine learning model fits the training data. It quantifies the difference between the predicted outputs by the model and the actual target values, with lower values indicating a better fit. The training loss is computed during the training phase and is used to update the model's parameters to improve its performance.

3.1.18 Training Accuracy:

Training accuracy is a metric that measures the proportion of correctly predicted instances out of the total instances in the training dataset. It reveals about how well the machine learning model is performing on the data it was trained on. Higher training accuracy indicates that the model is effectively learning patterns in the training data.

3.1.19 Validation Accuracy:

Validation accuracy is a metric that measures the proportion of correctly predicted instances out of the total instances in the validation dataset. It indicates how well the machine learning model generalizes to new, unseen data and is used to evaluate the model's performance during training without bias from the training data. Higher validation accuracy suggests better model generalization.

CHAPTER 4

DISCUSSION AND EXPERIMENTAL RESULTS

4.1 Introduction:

The consistency of the suggested model is graded from one to four. Of the four models, Xception performs best on all metrics, including accuracy (99.75%), precision (0.9971), recall (0.9977), and F1 score (0.9974). The MobileNetV2V2 model, which has an F1 score of 0.9946, recall of 0.9941, precision of 0.9953, and accuracy of 99.50%, comes in second. With 98.50% accuracy, 0.9852 precision, 0.9844 recall, and 0.9846 F1 score, InceptionV3 comes in third. The accuracy of the final model VGG-16 was 97.01% and precision 0.9728, recall 0.9660 and F-1 score 0.9668 respectively.

4.2 InceptionV3 Model Experimental Results and Discussion:

Parameter:

Total params: 48356394

Trainable params: 26451210

Non-trainable params: 21905184

The InceptionV3 model has a total of 48,356,394 parameters. These parameters are categorized into two types: trainable and non-trainable parameters.

Trainable parameter, which are 26,451,210 in number are the parts of the model that the learning process adjusts to improve performance. These include weights and biases within the model that are updated during training to minimize errors.

Non-trainable parameters which are 21,905,184 in number are also the parts of the model that remain fixed during training. These might include pre-set features or settings that the model uses but does not adjust based on the training data. Trainable parameters are used to learn from the data and improve the model's predictions, while non-trainable parameters are used to keep certain parts of the model stable and consistent throughout the training process.

4.2.1 Accuracy:

Accuracy is used to determine how much the model is effective to classify fresh and rotten vegetables separately. The precision is stated for each epoch. Increasing the number of epochs enhances accuracy in this scenario.

```
Epoch 1/10
101/101 [==============================] - 919s 9s/step - loss: 6.4821 - accuracy: 0.7481 - val_loss: 1.4039 - val_accuracy: 0.9450
Epoch 2/10
101/101 [==============================] - 340s 3s/step - loss: 3.7638 - accuracy: 0.8880 - val_loss: 0.6142 - val_accuracy: 0.9825
Epoch 3/10
101/101 [==============================] - 334s 3s/step - loss: 2.4973 - accuracy: 0.9264 - val_loss: 1.1492 - val_accuracy: 0.9800
Epoch 4/10
101/101 [==============================] - 339s 3s/step - loss: 2.0685 - accuracy: 0.9505 - val_loss: 0.0817 - val_accuracy: 0.9950
Epoch 5/10
101/101 [==============================] - 337s 3s/step - loss: 2.1156 - accuracy: 0.9462 - val_loss: 0.8792 - val_accuracy: 0.9800
Epoch 6/10
101/101 [==============================] - 301s 3s/step - loss: 2.7019 - accuracy: 0.9499 - val_loss: 0.5274 - val_accuracy: 0.9875
Epoch 7/10
101/101 [==============================] - 336s 3s/step - loss: 2.2939 - accuracy: 0.9573 - val_loss: 0.1635 - val_accuracy: 0.9950
Epoch 8/10
101/101 [==============================] - 334s 3s/step - loss: 2.0927 - accuracy: 0.9623 - val_loss: 0.2448 - val_accuracy: 0.9925
Epoch 9/10
101/101 [==============================] - 336s 3s/step - loss: 2.0599 - accuracy: 0.9635 - val_loss: 0.4790 - val_accuracy: 0.9875
Epoch 10/10
101/101 [==============================] - 334s 3s/step - loss: 1.5969 - accuracy: 0.9678 - val_loss: 0.4493 - val_accuracy: 0.9850
```

Fig.4.2.1.1 Epoch run to get the highest accuracy of InceptionV3

The above figure is about the training process of InceptionV3 model across 10 cycles, known as epochs. Each epoch involves processing the entire training dataset once, and the figure shows the model's performance in terms of accuracy and loss for both the training and validation datasets. In the first epoch, the model starts with a high error rate (loss) and lower accuracy. As the training progresses through each epoch, the model adjusts its internal parameters to better understand the patterns in the data. This results in a decrease in the error rate and an increase in accuracy. By the second epoch, there is a noticeable improvement as the loss decreases significantly and the accuracy increases. This trend continues with each epoch, showing steady progress. The model's training loss decreases from 6.4821 in the first epoch to 1.5969 by the tenth epoch. Similarly, the training accuracy improves from 74.81% to 96.78%.

The validation data, which is used to check the performance of the model on unseen data, also shows a decrease in loss and an increase in accuracy, though there are slight fluctuations. Initially, the validation loss is 1.4039 and accuracy is 94.50%, which improves to 0.4493 loss and 98.50% accuracy by the tenth epoch.

4.2.2 Confusion Matrix:

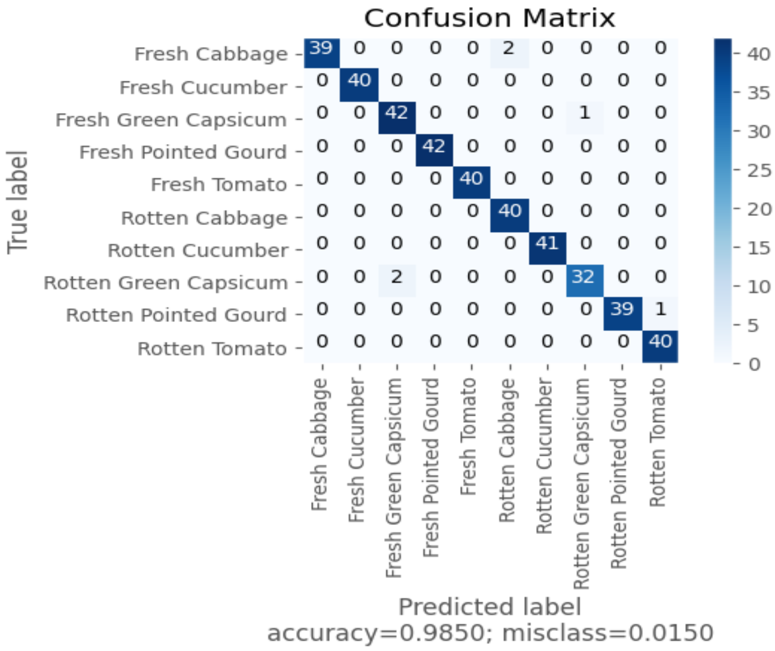

Figure 4.2.2.1 Confusion Matrix of InceptionV3.

In the above confusion matrix of InceptionV3 model, we can see that, for fresh cabbage out of 39 images, 39 images are predicted correct. In case of fresh cucumber out of 40 images, 40 images are also predicted correct. For fresh green capsicum out of 42 images, 41 images are predicted correct whereas 1 image is predicted wrong. For fresh pointed gourd, out of 42 images, 42 images are predicted correct. In case of fresh tomato, out of 40 images, 40 images are predicted correct. Now for rotten cabbage, out of 40 images, no picture is predicted wrong and so all pictures are predicted correct. For rotten cucumber, total 41 images are predicted correct and no wrong picture is found after prediction. Similarly, for rotten green capsicum, out of 32 images, 30 images are predicted correct and 2 images are predicted wrong. For rotten pointed gourd, out of 39 images, 38 images are predicted correct and 1 image is predicted wrong. For rotten tomato, all 40 images are predicted correct. The accuracy is at the rate of 98.5%.

4.2.3 Classification Performance metrics

The table shows the classification performance metrics for different categories of vegetables, both fresh and rotten. Precision, recall, and F1-score are listed for each category along with their support values, which indicate the number of instances in each class. The overall accuracy of the model is 0.9850, with macro and weighted averages for precision, recall, and F1-score being around 0.9852 to 0.9854. Fresh Cucumber and Fresh Pointed Gourd have perfect precision, recall, and F1-scores of 1.000, indicating the model perfectly classified all instances in these categories. Fresh Green Capsicum and Rotten Green Capsicum have slightly lower precision and recall compared to other categories but still maintain high F1-scores, indicating good overall performance.

	precision	recall	f1-score	support
Fresh Cabbage	1.0000	0.9512	0.9750	41
Fresh Cucumber	1.0000	1.0000	1.0000	40
Fresh Green Capsicum	0.9545	0.9767	0.9655	43
Fresh Pointed Gourd	1.0000	1.0000	1.0000	42
Fresh Tomato	1.0000	1.0000	1.0000	40
Rotten Cabbage	0.9524	1.0000	0.9756	40
Rotten Cucumber	1.0000	1.0000	1.0000	41
Rotten Green Capsicum	0.9697	0.9412	0.9552	34
Rotten Pointed Gourd	1.0000	0.9750	0.9873	40
Rotten Tomato	0.9756	1.0000	0.9877	40
accuracy			0.9850	401
macro avg	0.9852	0.9844	0.9846	401
weighted avg	0.9854	0.9850	0.9850	401

Figure 4.2.3.1 Classification Performance metrics of InceptionV3

Between the vegetables classes, precision, recall, f1-scrore, and accuracy are shown. With this model, we were able to achieve 98.50% accuracy rate.

4.2.4 Training Loss and Validation Loss:

Figure 4.2.3.1: Training Loss and Validation Loss of InceptionV3.

The graph shows the model's training and validation loss over ten epochs. The training loss (red line) decreases steadily at the beginning, indicating that the model is learning and fitting the training data better. The validation loss (blue line) also decreases initially but shows some fluctuations. By the end of the ten epochs, both losses have decreased, but the validation loss has stabilized, suggesting the model is not overfitting significantly. Overall, the model is improving in performance, with both training and validation losses reducing, though the fluctuations in validation loss suggest room for further improvement or tuning.

4.2.5 Training Accuracy and Validation Accuracy:

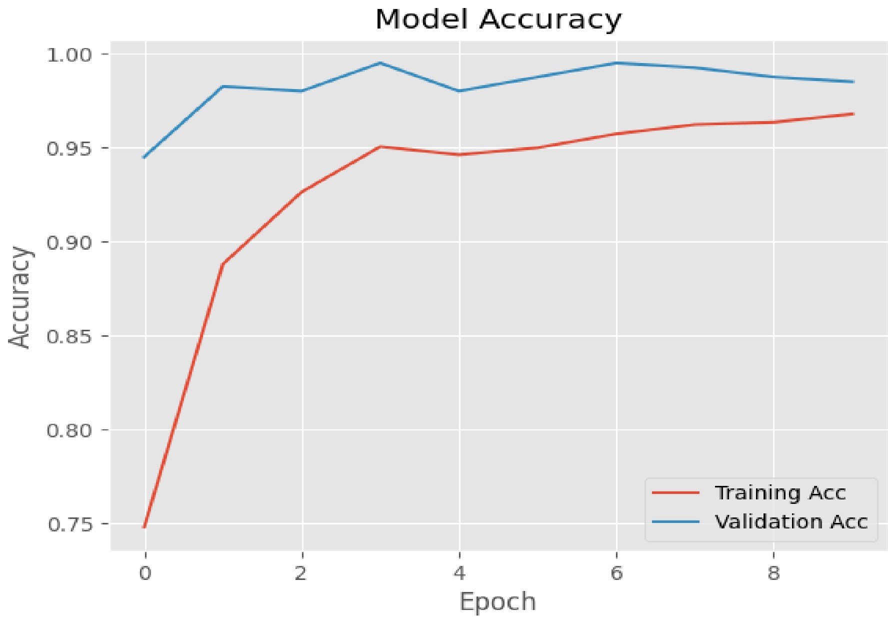

Figure 4.2.5.1: Training Accuracy and Validation Accuracy for InceptionV3.

The graph shows the model's training and validation accuracy over ten epochs. The training accuracy (red line) increases steadily, indicating the model is learning and improving its performance on the training data. The validation accuracy (blue line) starts high and shows minor fluctuations but remains relatively stable above 0.95. This suggests that the model is generalizing well to unseen data and not overfitting significantly. Overall, both training and validation accuracies improve, indicating a successful training process.

4.3 MobileNetV2 Model Experimental Results and Discussion:

4.3.1 Parameter:

Total params: 34755914

Trainable params: 32372490

Non-trainable params: 2383424

The total number of parameters of the model MobileNetV2 is 34,755,914. This number represents all the parameters in the model that contribute to its behavior and performance. Parameters are the values that the model learns during training to make accurate predictions.

Out of the total parameters, 32,372,490 are trainable. Trainable parameters are the ones that the model adjusts and updates during training. These parameters are essential for learning the patterns in the data and improving the model's accuracy.The remaining 2,383,424 parameters are non-trainable. Non-trainable parameters are usually part of pre-trained layers or fixed during the training process. These parameters do not get updated during training but still play a crucial role in the model's functionality.

4.3.2 Accuracy:

From accuracy we come to know about how much the model is effective to identify fresh and rotten vegetables separately. The precision is mentioned for each epoch. Accuracy in the scenario increases with the increase of the number of epochs.

```
Epoch 1/10
101/101 [==============================] - 819s 8s/step - loss: 3.4121 - accuracy: 0.8051 - val_loss: 0.9291 - val_accuracy: 0.9475
Epoch 2/10
101/101 [==============================] - 123s 1s/step - loss: 2.2571 - accuracy: 0.9202 - val_loss: 0.4139 - val_accuracy: 0.9775
Epoch 3/10
101/101 [==============================] - 127s 1s/step - loss: 1.4884 - accuracy: 0.9418 - val_loss: 0.0016 - val_accuracy: 1.0000
Epoch 4/10
101/101 [==============================] - 133s 1s/step - loss: 1.2378 - accuracy: 0.9610 - val_loss: 1.4729e-05 - val_accuracy: 1.0000
Epoch 5/10
101/101 [==============================] - 124s 1s/step - loss: 1.6825 - accuracy: 0.9536 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 6/10
101/101 [==============================] - 122s 1s/step - loss: 1.1232 - accuracy: 0.9623 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 7/10
101/101 [==============================] - 122s 1s/step - loss: 1.1979 - accuracy: 0.9684 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 8/10
101/101 [==============================] - 131s 1s/step - loss: 1.4518 - accuracy: 0.9684 - val_loss: 0.0000e+00 - val_accuracy: 1.0000
Epoch 9/10
101/101 [==============================] - 131s 1s/step - loss: 0.8552 - accuracy: 0.9790 - val_loss: 1.4595e-05 - val_accuracy: 1.0000
Epoch 10/10
101/101 [==============================] - 119s 1s/step - loss: 1.2290 - accuracy: 0.9765 - val_loss: 0.2101 - val_accuracy: 0.9950
```

Fig.4.3.2.1 Epoch run to get the highest accuracy of MobileNetV2

This figure shows how deep learning model MobileNetV2 improves over 10 training cycles, called epochs. Each epoch processes the entire training data, and we see the time it takes, how well the model learns from the training data, and how well it performs on separate validation data. Initially, the model has a high error rate, but as training progresses, it makes fewer mistakes and becomes more accurate, eventually performing almost perfectly.

4.3.3 Classification Performance metrics

	precision	recall	f1-score	support
Fresh Cabbage	1.0000	1.0000	1.0000	41
Fresh Cucumber	1.0000	1.0000	1.0000	40
Fresh Green Capsicum	0.9773	1.0000	0.9885	43
Fresh Pointed Gourd	1.0000	1.0000	1.0000	42
Fresh Tomato	1.0000	1.0000	1.0000	40
Rotten Cabbage	1.0000	1.0000	1.0000	40
Rotten Cucumber	1.0000	1.0000	1.0000	41
Rotten Green Capsicum	1.0000	0.9412	0.9697	34
Rotten Pointed Gourd	1.0000	1.0000	1.0000	40
Rotten Tomato	0.9756	1.0000	0.9877	40
accuracy			0.9950	401
macro avg	0.9953	0.9941	0.9946	401
weighted avg	0.9951	0.9950	0.9950	401

Figure 4.3.3.1 Classification Performance metrics of MobileNetV2

The table is about the classification performance metrics of MobileNetV2 that shows performance of a classification model on different types of vegetables, both fresh and rotten.

The performance of model is evaluated using precision, recall, and F1-score, along with the number of samples for each category (support). Precision measures the accuracy of the positive predictions. Recall measures the ability of the model to identify all positive instances. F1-score is the harmonic mean of precision and recall, providing a balance between the precision and recall. The model perfectly identified fresh cabbage, fresh cucumber, fresh pointed gourd, and fresh tomato, achieving a precision, recall, and F1-score of 1.000 each. The model also performed very well on the other categories, with slightly lower but still high scores for fresh green capsicum and rotten green capsicum. The overall accuracy of the model is 99.5%, with the macro and weighted averages of the metrics being very close to this value, indicating a consistently high performance across all categories.

4.3.4 Confusion Matrix:

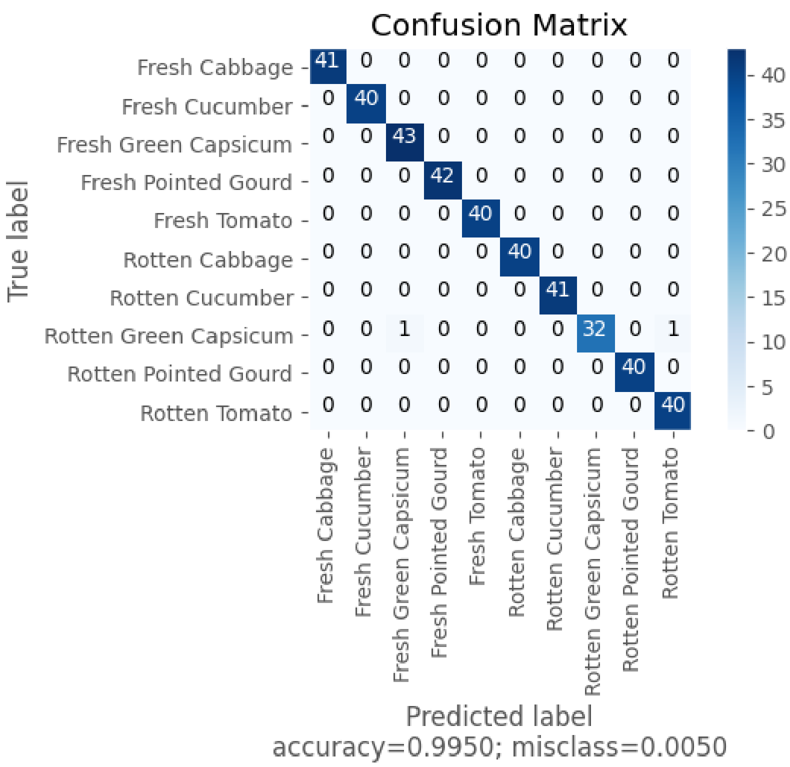

Figure 4.3.4.1 Confusion Matrix of MobileNetV2

In the above confusion matrix of MobileNetV2 model, we can see that, for fresh cabbage out of 41 images, 41 images are predicted correct.

In case of fresh cucumber out of 40 images, 40 images are also predicted correct. For fresh green capsicum out of 43 images, 43 images are predicted correct. For fresh pointed gourd, out of 42 images, 42 images are predicted correct. In case of fresh tomato, out of 40 images, 40 images are predicted correct. Now for rotten cabbage, out of 40 images, no picture is predicted wrong and so all pictures are predicted correct. For rotten cucumber, total 41 images are predicted correct and no wrong picture is found after prediction. Similarly, for rotten green capsicum, out of 32 images, 31 images are predicted correct and 1 image is predicted wrong. For rotten pointed gourd, out of 40 images, all 40 images are predicted correct. For rotten tomato, all 40 images are predicted correct. The accuracy is at the rate of 99.5%.

4.3.5 Training Loss and Validation Loss:

Figure 4.3.5.1: Training Loss and Validation Loss for MobileNetV2.

This graph shows how much error or "loss" the model has as it learns from data over 10 rounds, called epochs. The x-axis at the bottom shows the number of epochs, and the y-axis on the left shows the amount of loss. The red line represents training loss, which shows the error when the model is learning from its own data. It starts high and decreases quickly, but fluctuates a bit after a few epochs, ending around 1.0. The blue line represents validation loss, which shows the error on new, unseen data. It starts lower and quickly drops to almost zero, staying very low with a slight increase at the end. This means the model quickly improves and makes fewer mistakes on new data as it learns.

4.3.6 Training Accuracy and Validation Accuracy:

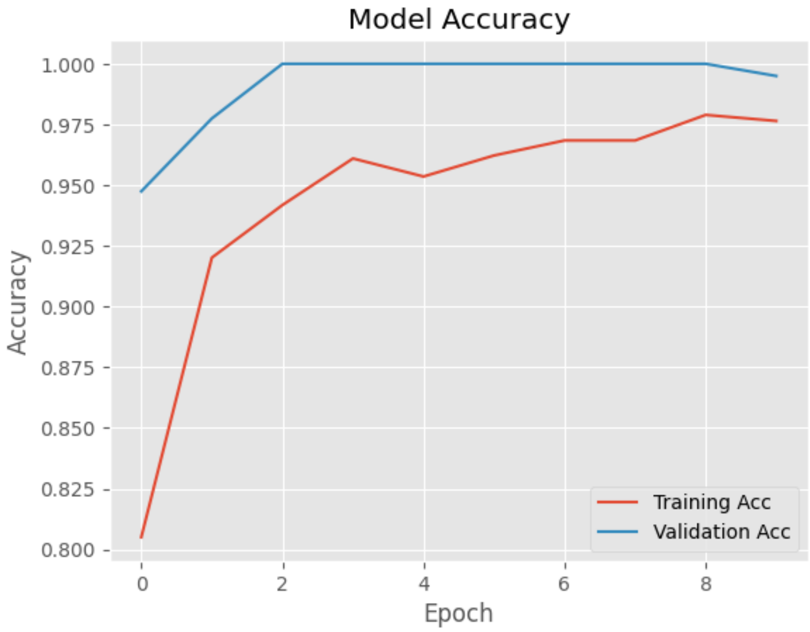

Figure 4.3.6.1: Training Accuracy and Validation Accuracy for MobileNetV2.

This graph shows how well a model MobileNetV2 is performing as it learns from data over 10 rounds, called epochs. The x-axis at the bottom shows the number of epochs, and the y-axis on the left shows accuracy, or how often the model gets the right answer. The red line is for training accuracy, showing how well the model is doing on the data it learns from directly. It starts lower but gets better quickly, leveling off around 95%. The blue line is for validation accuracy, showing how well the model does on new, unseen data. This line starts high and stays very close to perfect, around 100%, with only slight changes. This means the model is doing a great job on new data right from the start.

4.4 VGG-16 Model Experimental Results and Discussion:

4.4.1 Parameter:

Total params: 27794506

Trainable params: 13029642

Non-trainable params: 14764864

The model has a total of 27,794,506 parameters, which are the components that the model adjusts during training to make accurate predictions. Out of these, 13,029,642 parameters can be trained by the model during the learning process. The remaining 14,764,864 parameters are fixed and cannot be changed during training; they might be pre-set values or part of layers that do not update during the training phase. This mix of trainable and non-trainable parameters helps the model balance flexibility and stability.

4.4.2 Accuracy:

From accuracy it can be understood about how much the model is appropriate to give properly the feedback about which vegetables are fresh and rotten vegetables separately. The precision is illustrated for each epoch. Accuracy in the scenario increases with the increase of the number of epochs.

```
Epoch 1/10
101/101 [==============================] - 673s 7s/step - loss: 3.3456 - accuracy: 0.5662 - val_loss: 0.4893 - val_accuracy: 0.8550
Epoch 2/10
101/101 [==============================] - 39s 381ms/step - loss: 1.8736 - accuracy: 0.7432 - val_loss: 0.2234 - val_accuracy: 0.9350
Epoch 3/10
101/101 [==============================] - 39s 379ms/step - loss: 1.1243 - accuracy: 0.8150 - val_loss: 0.1108 - val_accuracy: 0.9525
Epoch 4/10
101/101 [==============================] - 39s 390ms/step - loss: 1.0103 - accuracy: 0.8292 - val_loss: 0.1103 - val_accuracy: 0.9550
Epoch 5/10
101/101 [==============================] - 40s 397ms/step - loss: 0.7734 - accuracy: 0.8620 - val_loss: 0.0709 - val_accuracy: 0.9825
Epoch 6/10
101/101 [==============================] - 41s 403ms/step - loss: 0.6513 - accuracy: 0.8663 - val_loss: 0.1211 - val_accuracy: 0.9500
Epoch 7/10
101/101 [==============================] - 39s 387ms/step - loss: 0.5501 - accuracy: 0.8775 - val_loss: 0.0659 - val_accuracy: 0.9700
Epoch 8/10
101/101 [==============================] - 38s 381ms/step - loss: 0.5541 - accuracy: 0.8874 - val_loss: 0.0677 - val_accuracy: 0.9775
Epoch 9/10
101/101 [==============================] - 38s 379ms/step - loss: 0.4375 - accuracy: 0.8967 - val_loss: 0.0832 - val_accuracy: 0.9725
Epoch 10/10
101/101 [==============================] - 39s 388ms/step - loss: 0.4591 - accuracy: 0.8923 - val_loss: 0.0868 - val_accuracy: 0.9700
```

Fig.4.5.2.1: Epoch run to get the highest accuracy of VGG-16

This above figure shows the training process of the model VGG-16 over 10 epochs. Each line represents one epoch and includes information about how long the epoch took, the loss, and accuracy on both the training and validation data.

At the start during Epoch 1, the training loss is high (3.3456) and accuracy is low (0.5662), but the validation loss is low (0.4893) and accuracy is high (0.8550). As training progresses, the training loss decreases significantly, and accuracy improves, showing the model is learning from the training data.

The validation loss stays consistently low, and validation accuracy remains high, indicating the model is performing well on unseen data. By the last epoch (Epoch 10), the training loss is much lower (0.4591) and accuracy is higher (0.8923), while the validation metrics show that the model is still generalizing well to new data (low loss of 0.0868 and high accuracy of 0.9700).

4.4.3 Classification Performance metrics

	precision	recall	f1-score	support
Fresh Cabbage	1.0000	1.0000	1.0000	41
Fresh Cucumber	1.0000	1.0000	1.0000	40
Fresh Green Capsicum	1.0000	1.0000	1.0000	43
Fresh Pointed Gourd	0.9333	1.0000	0.9655	42
Fresh Tomato	1.0000	1.0000	1.0000	40
Rotten Cabbage	1.0000	1.0000	1.0000	40
Rotten Cucumber	1.0000	1.0000	1.0000	41
Rotten Green Capsicum	1.0000	0.7353	0.8475	34
Rotten Pointed Gourd	0.9250	0.9250	0.9250	40
Rotten Tomato	0.8696	1.0000	0.9302	40
accuracy			0.9701	401
macro avg	0.9728	0.9660	0.9668	401
weighted avg	0.9725	0.9701	0.9690	401

Figure 4.4.3.1: Classification Performance metrics of VGG-16

The table shows how well VGG-16 model performs in identifying different types of vegetables as either fresh or rotten. The performance metrics include precision (accuracy of positive predictions), recall (ability to identify all relevant cases), and F1-score (balance between precision and recall). The number of instances for each category is shown under support.

The model perfectly identifies fresh cabbage, fresh cucumber, fresh green capsicum, fresh tomato, rotten cabbage, and rotten cucumber and achieved scores of 1.000 for precision, recall, and F1-score. The model performs slightly less well on fresh pointed gourd, rotten green capsicum, rotten pointed gourd, and rotten tomato, with varying but generally high scores.

The model has an accuracy of 97.01%. So, it correctly predicts the type of vegetable most of the time. The macro and weighted averages for precision, recall, and F1-score are all around 97%, reflecting the model's consistent performance across different categories.

4.4.4 Confusion Matrix:

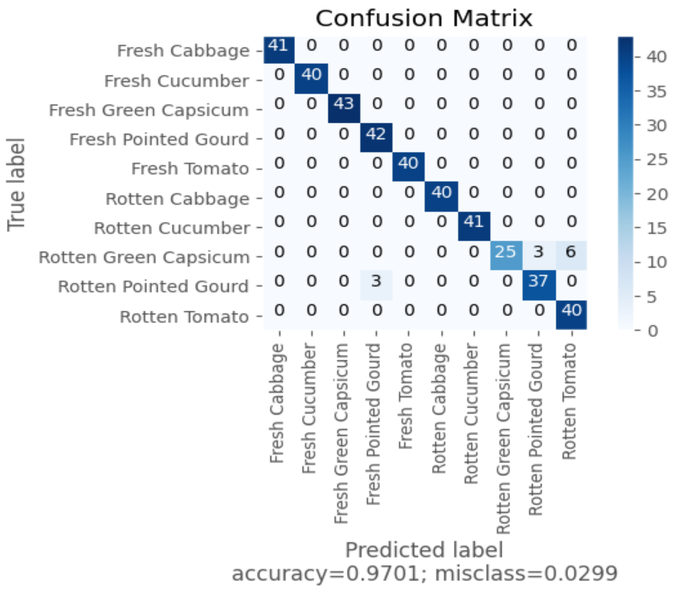

Figure 4.4.4.1: Confusion Matrix of VGG-16

In the above confusion matrix of VGG-16 model, we can see that, for fresh cabbage out of 41 images, 41 images are predicted correct. In case of fresh cucumber out of 40 images, 40 images are also predicted correct. For fresh green capsicum out of 43 images, 43 images are predicted correct. For fresh pointed gourd, out of 42 images, 42 images are predicted correct. In case of fresh tomato, out of 40 images, 40 images are predicted correct. Now for rotten cabbage, out of 40 images all pictures are predicted correct. For rotten cucumber, total 41 images are predicted correct and no incorrect picture is found after prediction.

Similarly, for rotten green capsicum, out of 25 images, 16 images are predicted correct and 9 images are predicted wrong. For rotten pointed gourd, out of 37 images, all 37 images are predicted correct. For rotten tomato, all 40 images are predicted correct. The accuracy is at the rate of 97.01%.

4.4.5 Training Loss and Validation Loss:

Figure 4.4.5.1: Training Loss and Validation Loss of VGG-16.

The graph shows how the error in the model's predictions changes over 10 training periods (epochs). The red line represents the error during training, while the blue line represents the error when the model is tested on new, unseen data.

At the beginning, the training error is high but quickly decreases, meaning the model is learning and improving. The error for the new data also decreases at first, showing the model is getting better at making accurate predictions. After a few training periods, both lines level off, indicating that the model's performance has stabilized and is not improving much further.

4.4.6 Training Accuracy and Validation Accuracy:

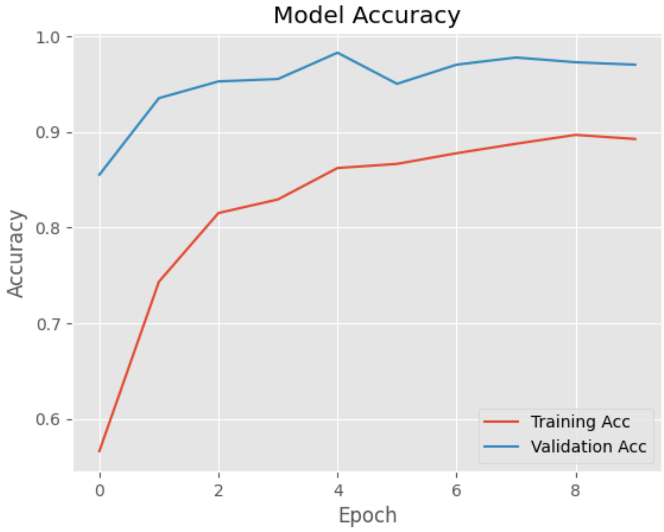

Figure 4.4.6.1: Training Accuracy and Validation Accuracy of VGG-16.

The graph shows how the accuracy of VGG-16 model changes over 10 training periods (epochs). The red line represents the accuracy during training, while the blue line represents the accuracy when the model is tested on new, unseen data.

At the start, the training accuracy is low but quickly improves ***i.e.*** the model is learning and making more correct predictions. The accuracy for the new data is higher from the beginning and also improves, showing that the model is good at predicting correctly on unseen data. After a few epochs, both lines level off, indicating that the model's performance has stabilized and is consistently accurate.

4.5 Xception Model Experimental Results and Discussion:

4.5.1 Parameter:

Total params: 72777522

Trainable params: 51715338

Non-trainable params: 21062184

The Xception model has a total of 72,777,522 parameters, which are the elements the model uses to make predictions. Out of these, 51,715,338 parameters can be adjusted during the training process, meaning the model can learn and improve by changing these values. The remaining 21,062,184 parameters are fixed and cannot be changed during training; these might include pre-set values or parameters that are part of layers that remain constant during the training. This combination of trainable and non-trainable parameters helps the model balance learning flexibility and stability.

4.5.2 Accuracy:

Accuracy is helpful to let us understand about how much the model can be useful to classify fresh and rotten vegetables separately. The precision is found out for each epoch. Accuracy increases along with the increase of the number of epochs.

```
Epoch 1/10
101/101 [==============================] - 1209s 12s/step - loss: 14.6176 - accuracy: 0.7104 - val_loss: 0.3996 - val_accuracy: 0.9800
Epoch 2/10
101/101 [==============================] - 548s 5s/step - loss: 7.8072 - accuracy: 0.8886 - val_loss: 0.5789 - val_accuracy: 0.9800
Epoch 3/10
101/101 [==============================] - 608s 6s/step - loss: 7.1736 - accuracy: 0.9158 - val_loss: 0.4954 - val_accuracy: 0.9850
Epoch 4/10
101/101 [==============================] - 605s 6s/step - loss: 5.3002 - accuracy: 0.9288 - val_loss: 0.5504 - val_accuracy: 0.9900
Epoch 5/10
101/101 [==============================] - 542s 5s/step - loss: 4.7343 - accuracy: 0.9375 - val_loss: 1.2798 - val_accuracy: 0.9775
Epoch 6/10
101/101 [==============================] - 598s 6s/step - loss: 4.5041 - accuracy: 0.9493 - val_loss: 0.6646 - val_accuracy: 0.9775
Epoch 7/10
101/101 [==============================] - 541s 5s/step - loss: 3.1861 - accuracy: 0.9505 - val_loss: 0.0485 - val_accuracy: 0.9925
Epoch 8/10
101/101 [==============================] - 598s 6s/step - loss: 3.6934 - accuracy: 0.9592 - val_loss: 0.5480 - val_accuracy: 0.9875
Epoch 9/10
101/101 [==============================] - 540s 5s/step - loss: 2.7686 - accuracy: 0.9585 - val_loss: 0.0826 - val_accuracy: 0.9975
Epoch 10/10
101/101 [==============================] - 596s 6s/step - loss: 2.5184 - accuracy: 0.9678 - val_loss: 0.1049 - val_accuracy: 0.9975
```

Fig.4.5.2.1: Epoch run to get the highest accuracy of Xception

The figure shows the training progress of the model Xception over 10 epochs. Each line represents one epoch, displaying the time taken per step, the loss (error) during training, the accuracy during training, the validation loss, and the validation accuracy.

At the beginning which is Epoch 1, the training accuracy is 71.04% with a high loss, while the validation accuracy is 98%. As training progresses, both training accuracy and validation accuracy generally improve. By the last epoch (Epoch 10), the training accuracy reaches 96.78%, and the validation accuracy is very high at 99.75%. This indicates that the model is learning well and performing accurately on unseen data.

4.5.3 Classification Performance metrics

	precision	recall	f1-score	support
Fresh Cabbage	1.0000	1.0000	1.0000	41
Fresh Cucumber	1.0000	1.0000	1.0000	40
Fresh Green Capsicum	1.0000	0.9767	0.9882	43
Fresh Pointed Gourd	1.0000	1.0000	1.0000	42
Fresh Tomato	1.0000	1.0000	1.0000	40
Rotten Cabbage	1.0000	1.0000	1.0000	40
Rotten Cucumber	1.0000	1.0000	1.0000	41
Rotten Green Capsicum	0.9714	1.0000	0.9855	34
Rotten Pointed Gourd	1.0000	1.0000	1.0000	40
Rotten Tomato	1.0000	1.0000	1.0000	40
accuracy			0.9975	401
macro avg	0.9971	0.9977	0.9974	401
weighted avg	0.9976	0.9975	0.9975	401

Figure 4.5.3.1: Classification Performance metrics of Xception

The table of Classification Performance metrics shows how well the Xception model can identify different vegetables as fresh or rotten. The model is evaluated using precision (how accurate the predictions are), recall (how well the model identifies all relevant cases), and F1-score (a balance between precision and recall).

For most vegetables, the model has perfect scores of 1.000, meaning it correctly identifies them every time. However, for rotten green capsicum, the scores are slightly lower, but still very high. Overall, the model's accuracy is 99.75%, showing it performs very well in classifying the vegetables correctly.

4.5.4 Confusion Matrix:

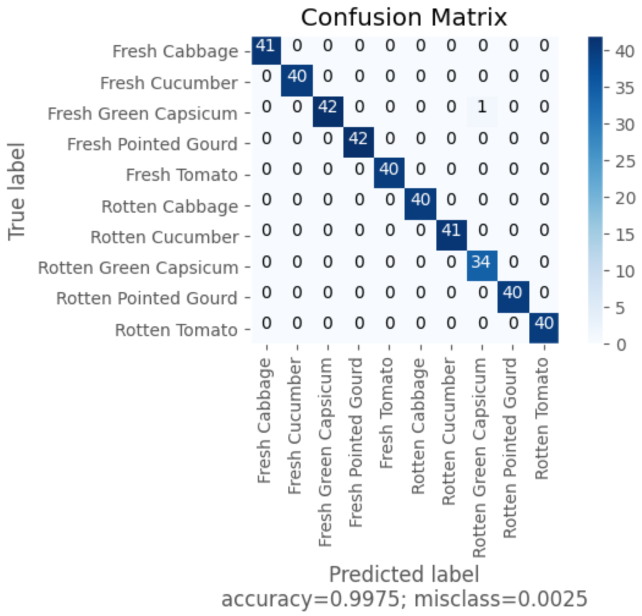

Fig.4.5.4.1: Confusion Matrix of Xception

In the above confusion matrix of Xception model, we can see that, for fresh cabbage out of 41 images, 41 images are predicted correct. In case of fresh cucumber out of 40 images, 40 images are also predicted correct. For fresh green capsicum out of 42 images, all the 42 images are predicted correct. For fresh pointed gourd, out of 42 images, all those 42 images are predicted correct. In case of fresh tomato, out of 40 images, 40 images are predicted correct. Now for rotten cabbage, out of 40 images all pictures are predicted correct. For rotten cucumber, total 41 images are predicted correct. Similarly, for rotten green capsicum, out of 34 images, all 34 images are predicted correct. For rotten pointed gourd, out of 40 images, all 40 images are predicted correct. For rotten tomato, all 40 images are predicted correct. Here it is found that no matter how many number of image data are shown, all image data are predicted correct. No image dataset were found which is predicted incorrect. The accuracy is at the rate of 97.01%.

4.5.5 Training Loss and Validation Loss:

Fig 4.5.5.1: Training Loss and Validation loss of Xception

The graph shows how the training loss (Red) and validation loss (Blue) of Xception model changed over 10 epochs. Training loss represents how well the model is learning from the training data, while validation loss indicates how well the model is performing on unseen data. Initially, the training loss is very high but decreases rapidly, indicating the model is learning quickly. However, the validation loss remains low and relatively stable, showing that the model's performance on new data doesn't improve much and stays consistent. This suggests the model is likely overfitting the training data, as the gap between training and validation loss remains large.

4.5.6 Training Accuracy and Validation Accuracy:

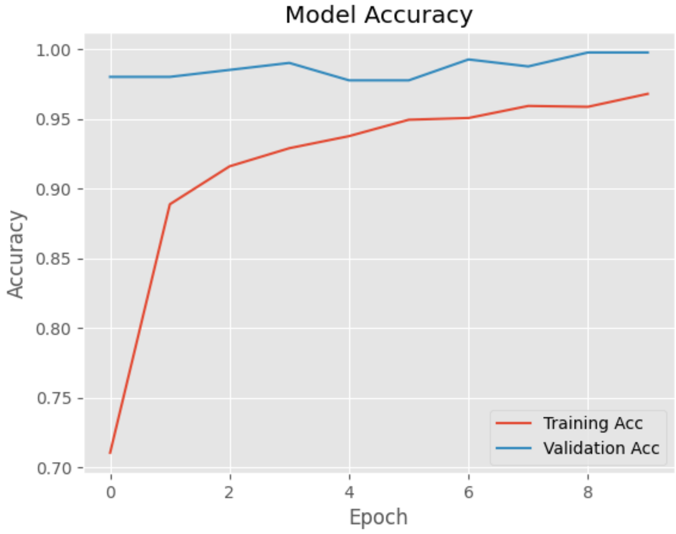

Fig 4.5.6.1 Training Accuracy and Validation Accuracy of Xception

This graph shows the accuracy of the model on both the training data and the validation data over 10 epochs. Accuracy measures how many predictions the model got right. At the beginning, the training accuracy is low, but it quickly improves, indicating the model is learning from the training data. The validation accuracy starts high and remains fairly consistent, slightly fluctuating but staying close to 100% accuracy. This suggests that while the model is getting better at predicting the training data correctly, its performance on new, unseen data remains consistently high. This means the model is likely performing well overall and not overfitting.

4.6 Discussion points:

Both the training and validation accuracy and loss of the InceptionV3 model are shown. Here it is found that the training accuracy was 74.81% in the first epoch, while the validation accuracy was 94.50%. In this way upto ten epochs, the validation accuracy was 98.50% and the training accuracy was 96.78%. The Confusion Matrix graphic shows that the InceptionV3 Model wrongly proved the predictions for six out of the 401 photos in the test set. For example, when it came to fresh cabbage, only two out of 41 photographs had wrong predictions, while 39 of the images had good predictions.

The study also shows the Xception Model's training accuracy and loss as well as its validation accuracy and loss. Here the training accuracy was 71% and the validation accuracy was 98% in the first epoch. The training accuracy was 96.78% and the validation accuracy was 99.75% after ten epochs. One of the 401 photos in the test set was inaccurately predicted by the Xception Model, as can be seen in the Confusion Matrix graphic. For instance, 42 of the 43 images of fresh green capsicum that were submitted had accurate forecasts, while only one had an incorrect one.

The diagram shows the training accuracy and loss of the VGG16 Model as well as the validation accuracy and loss. The training accuracy was 56.62% and the validation accuracy was 85.50% in the first period. After 10 epochs, the validation accuracy was 97% and the training accuracy was 89.23%. As can be seen in the Confusion Matrix picture, the VGG16 Model made inaccurate predictions for nine out of the forty-one photos in the test set. For instance, 25 of the 32 images of rotting green capsicum that were used for the prediction were accurate, whereas 9 were not.

The diagram displays the MobileNetV2 Model's training accuracy and loss as well as its validation accuracy and loss. The training accuracy was 80.51 percent and the validation accuracy was 94.75 percent in the first period. The training accuracy was 96.75% and the validation accuracy was 99.50% after ten epochs. One of the 401 photos in the test set that the MobileNetV2 Model erroneously predicted may be seen in the Confusion Matrix.

For instance, out of 33 pictures of rotting green capsicum, 32 were predicted correctly and 1 was predicted wrong.

The final results of the four models are shown below:

Model	Accuracy	Precision	Recall	F1 Score
InceptionV3	98.50%	0.9852	0.9844	0.9846
Xception	99.75%	0.9971	0.9977	0.9974
VGG-16	97.01%	0.9728	0.9960	0.9968
MobileNetV2	99.50%	0.9953	0.9941	0.9946

Table: 4.6.1 Final result

The final results of the four models in bar chart are shown below:

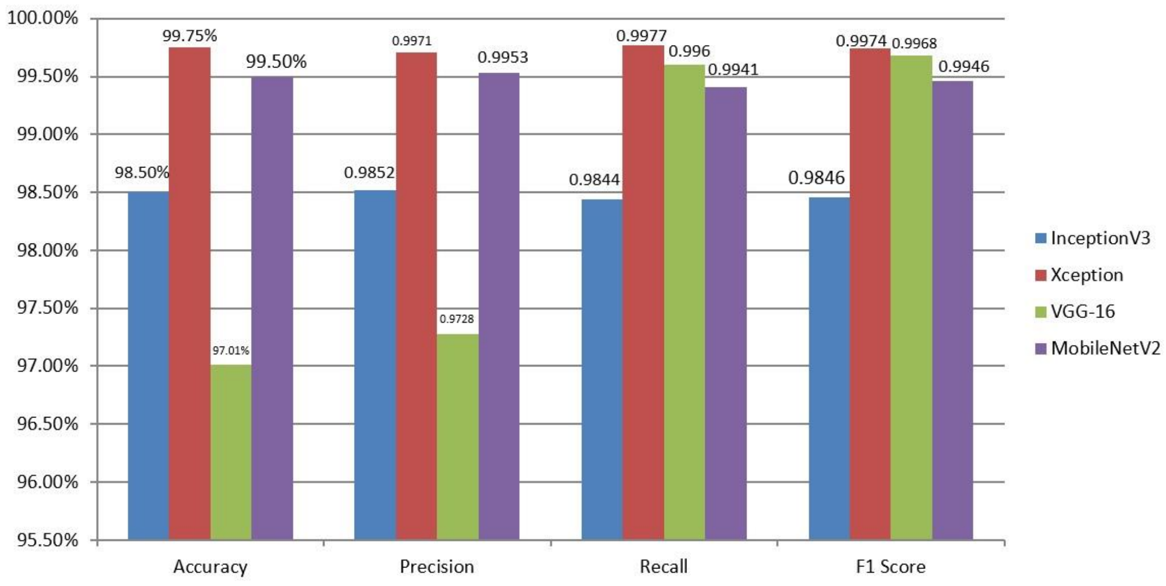

Figure: 4.6.1 Final result in bar chart

CHAPTER 5
SOCIETY'S IMPACT, SUSTAINABILITY AND THE ENVIRONMENT

5.1 Impact of society:

The impact of using deep learning to automatically detect fresh and rotten vegetables in society is immense. The architecture of CNN is being used in autonomous crop collection robots. Farmers can also use this method to cultivation of vegetables. This could result in consistent crop growth and additional harvests throughout the year. The problems of cultivating vegetable and the symptoms of vegetables getting rotten can be detected easily due to deep learning methods. As a result of this, the monotonous feelings of farmers will be removed and there will be no problems in sorting the fresh and decaying vegetables separately. The automated technology using the deep learning helps to find out the rotten free vegetables easily and correctly which reduces time. This will be helpful for farmers as they will be interested in increasing productions. The economy of the country will also be improved due to increasing productions of fresh vegetables correctly in less time and exporting in foreign countries.

This technology ensures that fresh vegetables will be consumed by the people in order to avoid the health risks. There will be less possibility of getting attacked by diseases. It improves the safety of food. Due to this technology, decision can be taken in a short period of time about the place where vegetables are to be kept and also how to keep the fresh vegetables in a safe place in order to avoid decaying. The trust of customers will also be increased in purchasing fresh vegetables from the market or groceries. For farmers and suppliers, it reduces waste by sorting vegetables accurately by ensuring that only the fresh vegetables can be sold, which also maximizes profits and will improve the standard of living of farmers and suppliers.

In the export market, maintaining high standards of vegetable freshness increases the reputation of exporters and supports economic growth of the country. Exporting fresh vegetables in the foreign countries will also help in increasing confidence of the exporters. So it can be said that this technology promotes better health, stability of economy, and efficiency in food production and distribution.

5.2 Impact on Environment

The implementation of detecting fresh and rotten vegetables using deep learning has a great impact on the environment in several ways. This technology can accurately reduce wastage of food. At present, a large amount of food including vegetables, is wasted because of handling improperly, storage, and late detection of getting rotten. By using deep learning, differences between fresh and rotten vegetables can be known quickly and accurately, food distributors and retailers can ensure that only fresh vegetables produced reaches to consumers.

Reducing wastage of food is very important because it has direct environmental benefits. When food is wasted, all the resources used to produce, process, transport, and store that food will also be in vain. The resources include water, energy, and labor. Moreover, decomposing food waste on land generates methane, a greenhouse gas that contributes to the change of climate. By minimizing wastage of food, the automated detection process helps in reducing the emission of greenhouse gases and conserves valuable resources.

Farmers and producers can receive real-time feedback about the quality of their vegetable production which allows them to make informed decisions about harvesting and post-harvest handling and this can lead to the adoption of better farming techniques that reduces crop losses and improve quality of yield, ultimately promoting sustainable agriculture.

The automated detection system helps local and regional food markets by making sure consumers get fresh, high-quality vegetables. This promotes eating locally grown produce and reduces the need for importing vegetables from far away, which is better for the environment due to less long-distance transportation. The exporting of local fresh vegetables can be ensured by the help of automated detection system. This will help in improving the economic condition of the country. The farmers of the country will also be benefitted through the profit and they will be more interested in growing vegetables. The exporters will not be frustrated at selling the vegetables in the foreign countries. Exporting more fresh vegetables in the foreign countries will contribute to the development of the economic condition of the country. The country will automatically develop in economy due to this.

5.3 Ethical Aspects

Classification methods can be utilized without doing any damage to the environment, and environment can be benefitted much. Vegetable classification may lead to increase of crop yields and less wastage by removing the need for vegetable preservation medicine. Ensuring that the technology is developed, classifying fresh and rotten vegetables reduces time wastage and help in increase of vegetable production. There is the issue of data privacy and security. Deep learning systems rely on large amounts of functional data effectively. This data might include images and information about vegetables from various farms and suppliers. If the data is collected, stored, and processed in a manner that detection of fresh and rotten vegetables will be easier and fresh and decayed portions will be separated. Unauthorized access or misuse of this data could lead to privacy violations or disadvantages in competition for suppliers. Besides, there is the possibility of job replacement. The introduction of automated systems can lead to the reduction of jobs traditionally done by humans, such as quality control inspectors in food processing and distribution centers. While automation of vegetable detection create new opportunities in tech development and maintenance, it is essential to address the potential job losses and provide training or alternative employment opportunities for those affected. Additionally, there is the question of fairness and accessibility. The benefits of this technology should be accessible to all, including small-scale farmers and producers who might not have the resources to invest in advanced technology. While the system aims to cut down on food wastage and promote sustainability, it’s important that the production and disposal of its parts are environmentally friendly. Disposing the rotten vegetables will also increase the fertility of the soil and more crops and vegetables can be grown. The accuracy and reliability of the deep learning model are critical. If the system incorrectly identifies fresh vegetables as rotten or vice versa, it could lead to unnecessary waste or health risks. Ensuring the technology is thoroughly correctness of detection is essential to avoid such errors and maintaining trust among consumers and producers. Users of the system should be informed about how the technology works, what data it uses, and how decisions are made. This transparency helps build trust and allows users to make informed decisions about its use.

5.4 Sustainability Plan

Smart mobile applications can be created to identify the vegetables swiftly in a certain location or to collect critical information such as nutritional values. These apps can also be used to maintain track of inventory at supermarkets.

The sustainability plan for an automated process of detecting fresh and rotten vegetables using deep learning involves several aspects to ensure the technology is both effective and environmentally friendly. By accurately identifying which vegetables are fresh and which are rotten, we can make sure that only good quality vegetables are required in different localities. Throwing fewer vegetables will help in conserving resources like water, energy, and labor that go into growing and transporting food. Less food wastage also means less organic wastage in landfills, which can reduce the production of harmful greenhouse gases like methane. The technology should be accessible to all farmers and producers. This helps promote equity in the agricultural sector by allowing the farmers to get benefit from the same efficiencies and sustainability practices as larger operations.

By providing support and training, we can help these farmers habituated in using the technology to avoid wastage of food, encouraging them to grow more and more vegetables for the betterment of the society, country and throughout the world. This technology has to be handed over the farmers in low-cost.

CHAPTER 6
CONCLUSION, RECOMMENDATION, CONSEQUENCES FOR FUTURE RESEARCH

6.1 Conclusions:

In our study effort, we analyzed and assessed four suitable deep learning models, including InceptionV3, MobileNetV2V2, VGG-16, and Xception, for recognizing vegetables available in Bangladesh. We also discussed how they work and what applications they offer. These models are found accurate by us among all other models. For testing, we developed our own dataset of five distinct types of vegetables. The vegetables are classified into 10 classes. Preparing the data properly, we trained and evaluated each of the four models. Our goal was to determine how efficiently and accurately each of the four models classified fresh and rotten vegetables and it was performed successfully. Each of the four models InceptionV3, MobileNetV2V2, VGG-16, and Xception showed proper accuracy exceeding over 95% which is good for detection. The scope of this study will expand in the future to include more vegetables varieties for classification by allowing any vegetables farmer to apply the procedure. The proposed approach is more advantageous for vegetables growers in terms of distinguishing fresh and rotting vegetables based on yield in order to earn a higher selling price. The fresh and rotten vegetables detection using deep learning is useful to find out which vegetables are fresh and which are rotten in a quickest possible of time where manual labor is eliminated. It will also help in improving the condition of the economy of the country in shortest possible of time by the increase of selling price due to exporting in foreign countries. Dependency on importing from the foreign countries will be reduced. The family condition of farmers will be improved as they will get profit from the sales of fresh vegetables. There will be less possibility of getting attacked by diseases and the health will remain safe after consuming. The application of medicine in preserving the vegetables will be removed due to this. The corruption will be removed from the countries. Fraud will also be removed. Living in the society peacefully will be easier.

6.2 Recommendations:

The output of the deep learning models discussed here is depending on a number of variables, but all four models can detect vegetables. Xception is better model based on the accuracy and consistency of our own dataset. Because the Xception give the highest accuracy of 99.75% among all other models. It showed the fresh and rotten vegetables very accurately.

The Xception model is a type of deep learning neural network useful for the image recognition tasks. Xception learns to understand the images by breaking down the process into smaller, manageable parts. It starts by identifying basic elements like edges and corners and then it combines these to recognize more complex features such as shapes and textures. This approach not only improves accuracy but also speeds up training and inference by making it useful for tasks like object detection, image classification, and even medical image analysis.

It uses a two-step process to analyze the image pieces. First, it quickly checks the general details of each piece. Then, it zooms in on those pieces that seem interesting and analyzes them more carefully. Xception model breaks down the image into tiny pieces and examining them closely. Then, it puts the pieces back together with a newfound understanding. Xception represents a significant advancement in deep learning models, offering both efficiency and effectiveness in handling complex visual data. Its innovative design has contributed to its widespread adoption across various fields where image understanding is crucial. So Xception model is much useful among all other models.

6.3 Involvement for Future Research:

In future studies, we can include more types of vegetables and focus on classifying them into specific sub-categories. Using larger datasets has proven effective in deep convolutional neural networks (DCNNs), as they consistently achieve high accuracy rates. These models can be trained to recognize vegetables accurately in any picture.

We should also explore how these models perform under real-world challenges, like different lighting conditions or crowded settings where vegetables are stored or displayed. This will require further research into how convolutional neural networks handle these kinds of situations.

We recommend expanding the types of vegetables in the dataset to include a wider variety. This can help the deep learning model better recognize different vegetables and their specific characteristics of freshness or defect.

We also propose incorporating additional types of data, like smell or texture information, alongside visual data. This could enhance the accuracy of the model's predictions by providing more comprehensive information about vegetable quality.

Furthermore, the study suggests refining the model's algorithms and testing them under various real-world conditions. This could involve different lighting conditions or environments typical of food processing facilities.

Overall, future research aims to build on the current findings to create more suitable and reliable automated systems for assessing vegetable quality. This can potentially improve food safety and reduce waste in the food industry.

REFERENCES

[1] Roul, A., Padhy, S., Parhi, M., & Pati, A. (2023, June). Veggiefreshai: An intelligent system to detect vegetable quality using deep computational analysis. In *AIP Conference Proceedings* (Vol. 2819, No. 1). AIP Publishing.

[2] Akter, T., Mahamud, S., Iqbal, M., Swarna, T. A., Nabi, N., & Ahamed, M. S. (2022, December). Local Vegetable Freshness Classification Using Transfer Learning Approaches. In *2022 32nd International Conference on Computer Theory and Applications (ICCTA)* (pp. 90-95). IEEE.

[3] Yuan, Y., & Chen, X. (2024). Vegetable and fruit freshness detection based on deep features and principal component analysis. Current Research in Food Science, 8, 100656.

[4] Ahmed, R., Haque, M. M., Saha, S., Saha, C., Dutta, M., Karim, D. Z., ... & Rasel, A. A. (2023, July). Fresh or Stale: Leveraging Deep Learning to Detect Freshness of Vegetables and Vegetables. In 2023 3rd International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) (pp. 1-6). IEEE.

[5] Hamim, Md Abrar & Tahseen, Jeba & Hossain, Kazi Md. Istiyak & Akter, Nasrin & Asha, Umme. (2023). Bangladeshi Fresh-Rotten Fruit & Vegetable Detection Using Deep Learning Deployment in Effective Application. 233-238. 10.1109/CCAI57533.2023.10201244.

[6] Jana, S., Parekh, R., & Sarkar, B. (2021). Automated sorting of rotten or defective fruits and vegetables using convolutional neural network. In Proceedings of International Conference on Computational Intelligence, Data Science and Cloud Computing: IEM-ICDC 2020 (pp. 43-55). Springer Singapore.

[7] Zhou, C., Hu, J., Xu, Z., Yue, J., Ye, H., & Yang, G. (2020). A monitoring system for the segmentation and grading of broccoli head based on deep learning and neural networks. Frontiers in plant science, 11, 402.

[8] Li, Y., Xue, J., Wang, K., Zhang, M., & Li, Z. (2022). Surface defect detection of fresh-cut cauliflowers based on convolutional neural network with transfer learning. Foods, 11(18), 2915.

[9] Soltani Firouz, M., & Sardari, H. (2022). Defect detection in fruit and vegetables by using machine vision systems and image processing. Food Engineering Reviews, 14(3), 353-379.

[10] Du, Z., Zeng, X., Li, X., Ding, X., Cao, J., & Jiang, W. (2020). Recent advances in imaging techniques for bruise detection in Vegetables and vegetables. *Trends in Food Science & Technology*, *99*, 133-141.

[11] Baloch, A. K., Okatan, A., Kanasro, N. A., Baloch, M. A., & Jamali, A. A. (2023). The Quality Analysis of Food and Vegetable from Image Processing. *VAWKUM Transactions on Computer Sciences*, *11*(2), 01-17.

[12] Moreno-Mesonero, L., Soler, L., Amorós, I., Moreno, Y., Ferrús, M. A., & Alonso, J. L. (2023). Protozoan parasites and free-living amoebae contamination in organic leafy green vegetables and strawberries from Spain. Food and Waterborne Parasitology, 32, e00200.

[13] Jadhav, V. S. (2021). Development and advancement in image processing technique for detection of various diseases in Fruit & Vegetable Plants: A review. Int. J. Comput. Sci. Mob. Comput., 10(2), 31-38.

[14] Korchagin, S. A., Gataullin, S. T., Osipov, A. V., Smirnov, M. V., Suvorov, S. V., Serdechnyi, D. V., & Bublikov, K. V. (2021). Development of an optimal algorithm for detecting damaged and diseased potato tubers moving along a conveyor belt using computer vision systems. Agronomy, 11(10), 1980.

[15] Narendra, V. G., & Pinto, A. J. (2021). Defects detection in fruits and vegetables using image processing and soft computing techniques. In Proceedings of 6th International Conference on Harmony Search, Soft Computing and Applications: ICHSA 2020, Istanbul (pp. 325-337). Springer Singapore.

[16] VG, N., & Pinto, A. (2021). Defects Detection in Fruits and Vegetables Using Image Processing and Soft Computing Techniques. ResearchGate.

[17] Huynh, Q. K., Nguyen, C. N., Vo-Nguyen, H. P., Tran-Nguyen, P. L., Le, P. H., Le, D. K. L., & Nguyen, V. C. (2021). Crack identification on the fresh chilli (Capsicum) fruit destemmed system. Journal of Sensors, 2021(1), 8838247.

[18] B. L. Sirisha, B. J. Nayak, B. s. Sndhya, J. E. Rao and M. Devisri, "Fruit Quality Detection Using Deep Learning for Sustainable Agriculture," 2024 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, 2024, pp. 1-6, doi: 10.1109/RAEEUCCI61380.2024.10548005

[19] Singh, A., Gupta, R., & Kumar, A. (2023, March). Fresh and Rotten Fruit Detection Using Deep CNN and MobileNetV2. In International Conference on Advances in IoT and Security with AI (pp. 219-231). Singapore: Springer Nature Singapore.

[20] Reddy, M. S. S. A., & Aishwarya, N. (2023, April). A Deep Learning Approach to Identify Fresh and Stale Fruits and Vegetables with YOLO. In 2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS) (pp. 606-610). IEEE.

[21] Pazos, J. F. M., González, J. G., Lorenzo, D. B., & García, A. O. (2024). FRESHNets: Highly Accurate and Efficient Food Freshness Assessment Based on Deep Convolutional Neural Networks. Inteligencia Artificial, 27(74), 48-61.

[22] Valentino, F., Cenggoro, T. W., Elwirehardja, G. N., & Pardamean, B. (2023). Energy-efficient deep learning model for fruit freshness detection. Int J Artif Intell ISSN nd, 2252, 1387.

[23] Kumar, Y. R., Priya, V. V. L., Swathi, R., Priyanka, K., & Vyshnavi, G. (2023). Mushroom Decay Detection using Deep Learning. International Journal, 11(12).

[24] Hindarto, D. (2024). Model Performance Evaluation: VGG19 and Dense201 for Fresh Meat Detection. Sinkron: jurnal dan penelitian teknik informatika, 8(1), 514-524.

[25] Bu, Y., Hu, J., Chen, C., Bai, S., Chen, Z., Hu, T., ... & Gong, Y. (2024). ResNet incorporating the fusion data of RGB & hyperspectral images improves classification accuracy

of vegetable soybean freshness. Scientific Reports, 14(1), 2568.

[26] Sharmila, A., Tamilselvan, K. S., EL, D. P., Yuvaraj, D., Soundarya, B., & Ambika, K. S. (2023, August). Fruits Freshness Recognition System by Manipulating Versatile Neural Networks. In 2023 International Conference on Circuit Power and Computing Technologies (ICCPCT) (pp. 337-342). IEEE.

[27] Lin, Y., Ma, J., Sun, D. W., Cheng, J. H., & Zhou, C. (2024). Fast real-time monitoring of meat freshness based on fluorescent sensing array and deep learning: From development to deployment. Food chemistry, 448, 139078.

[28] Feng, J., Yang, Q., Tian, H., Wang, Z., Tian, S., & Xu, H. (2024). Promising real-time fruit and vegetable quality detection technologies applicable to manipulator picking process. International Journal of Agricultural and Biological Engineering, 17(2), 14-26.

[29] Evarist, N., Deborah, N., Birungi, G., Caroline, N. K., & Kule, B. J. M. (2024, January). A Model for Detecting the Presence of Pesticide Residues in Edible Parts of Tomatoes, Cabbages, Carrots, and Green Pepper Vegetables. In Artificial Intelligence and Applications (Vol. 2, No. 3, pp. 225-232).

[30] Senanayake, I. R., Weerasekara, B. J. D. A., Siriwardana, H. T. A., Ekanayake, N. G. R. P., Fernando, H., & Kelegama, T. (2023). Smart Navigation System for Supermarket. International Research Journal of Innovations in Engineering and Technology, 7(10), 543.

[31] Choudhury, T., Singh, T. P., Jain, P., Arunachalaeshwaran, V. R., & Sarkar, T. (2023). Radish Freshness Classification Using Deep Learning. In Intelligent Sustainable Systems: Selected Papers of WorldS4 2022, Volume 2 (pp. 483-493). Singapore: Springer Nature Singapore.

[32] Stasenko, N., Shadrin, D., Katrutsa, A., & Somov, A. (2023). Dynamic mode decomposition and deep learning for postharvest decay prediction in apples. IEEE Transactions on Instrumentation and Measurement, 72, 1-11.

[33] Sanyal, S., Adhikary, R., & Choudhury, S. J. (2024). Revolutionizing lemon grading: an automated CNN-based approach for enhanced quality assessment. International Journal of Information Technology, 1-12.

[34] Levaj, B., Pelaić, Z., Galić, K., Kurek, M., Ščetar, M., Poljak, M., ... & Repajić, M. (2023). Maintaining the quality and safety of fresh-cut potatoes (Solanum tuberosum): Overview of recent findings and approaches. Agronomy, 13(8), 2002.

[35] Al-Dairi, M., Pathare, P. B., & Al-Yahyai, R. (2024). Bruise Damage Susceptibility of Tomato. In Mechanical Damage in Fresh Horticultural Produce: Measurement, Analysis and Control (pp. 173-186). Singapore: Springer Nature Singapore.

An Automated Process of detecting fresh and rotten

ORIGINALITY REPORT

24%	19%	15%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	8%
2	Submitted to Daffodil International University Student Paper	2%
3	fastercapital.com Internet Source	1%
4	link.springer.com Internet Source	1%
5	www.mdpi.com Internet Source	1%
6	Submitted to University of Strathclyde Student Paper	1%
7	www.frontiersin.org Internet Source	<1%
8	Yue Yuan, Xianlong Chen. "Vegetable and fruit freshness detection based on deep features and principal component analysis", Current Research in Food Science, 2023 Publication	<1%