AN ADVANCED METHOD OF IDENTIFYING FRESH AND ROTTEN FRUITS USING DEEP LEARNING

BY

Md. Sohel Miah
ID: 171-15-9029

AND

Tayeeba Tasnuva
ID: 171-15-8704

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

Supervised by
Md. Riazur Rahman
Assistant Professor
Department of CSE
Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY

DHAKA, BANGLADESH

MAY 2021

APPROVAL

The title of this project/internship is **"An Advanced Method of Identifying Fresh and Rotten Fruits Using Deep Learning"**, submitted by Md. Sohel Miah, ID No: 171-15-9029 and Tayeeba Tasnuva, ID No: 171-15-8704. 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 its style and contents have been authorized by the Department of Computer Science and Engineering. The presentation took place in May of 2021

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan **Chairman**
Professor and Head
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Zahir

Gazi Zahirul Islam **Internal Examiner**
Assistant Professor
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

Raja Tariqul Hasan Tusher **Internal Examiner**
Senior Lecturer
Department of CSE
Jahangirnagar University

Dr. Dewan Md. Farid **External Examiner**
Associate Professor
Department of CSE
United International University

DECLARATION

We thus declare that we completed this study under the supervision of **Md. Riazur Rahman, Assistant Professor, Department of CSE** Daffodil International University. We further certify that no component of this project, or any component of it, has been submitted to any other institution for the granting of a degree or certificate.

Supervised by:

Md. Riazur Rahman
Assistant Professor
Department of CSE
Daffodil International University

Submitted by:

Md. Sohel Miah
ID: 171-15-9029
Department of CSE
Daffodil International University

Tayeeba Tasnuva

Tayeeba Tasnuva
ID: 171-15-8704
Department of CSE
Daffodil International University

ACKNOWLEDGEMENT

First of all, and foremost, we express our gratitude to Almighty God for His wonderful favor, which has enabled us to accomplish the final year project. **Md. Riazur Rahman,** Assistant Professor, Department of Computer Science and Engineering, Daffodil International University, Dhaka, deserves our gratitude and genuine admiration. To complete this project, our supervisor has extensive knowledge and a keen interest in the field of computer vision. This assignment was finished thanks to his unwavering patience, academic supervision, endless support, frequent and passionate oversight, suggestions for improvement, intelligent counsel, and reviewing numerous inferior drafts and revising them at every level.

Professor Dr. Touhid Bhuiyan, Professor and Head, Department of CSE, as well as the other academic members and employees of Daffodil International University's CSE department, deserve our sincerest gratitude for their aid in finishing our project. We are grateful to our Machine Learning Research Lab & Machine Learning Boot camp for providing us with instruments and providing us with the opportunity to complete this research.

Finally, we must respect and acknowledge our parents' continued support and patience.

ABSTRACT

Fruit classification is crucial in a variety of industries. This hierarchy aids sellers in various supermarkets in identifying fruit species and, as a result, impacts prices. The classification procedure makes it simple to decide whether the fruit is good or bad. People in the digital age take care of their everyday needs on their own, using a variety of contemporary facilities. If we are unable to export fresh fruits, our economic situation will deteriorate. In this example, a fruit categorization system could be useful in a range of fields, including the creation of smartphone apps for spotting unusual fruit species on the market and autonomous agricultural robots A total of 5658 fruits were used. (2834 fresh and 2824 rotten) in this study, which were divided into 10 classes (i.e 5 fresh fruits of 5 types & 5 rottenHumans become weary after performing the same role several times, so machines do not. As a result, the initiative suggests a plan for lowering costs and decreasing human effort. It is possible to reduce processing costs and time by detecting defects in agricultural fruits. We developed five models for our proposed classification system. The InceptionV3 model had the highest accuracy, at 97.34 percent.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Everyone's goal in life is to remain safe and healthy. It is critical to ensure sufficient nutrition as well as to focus on growth development and damage compensation in order to maintain good health. The majority of humans get their nutrients from fruits and vegetables. Fruits are high in fiber and contain a variety of vitamins and minerals. Fruits also contain a variety of antioxidants that are good for your health, such as flavonoids. Heart disease, cancer, inflammation, and diabetes can be avoided by eating a diet rich in fruits and vegetables. As a result, fresh fruits and vegetables are more likely to be consumed. Bangladesh is, as we all know, an agricultural nation. We will cultivate seeds of fruits and vegetables because our soil is fertile. Bangladesh's soil is ideal for growing fruits such as mango, banana, and plum since most of the country is rural. Despite being a tropical country, temperatures in Bangladesh range from 10 to 32 degrees Celsius throughout the year. Bangladesh has six separate seasons, each with plenty of rain. Bangladesh's soil is exceptionally fertile, making it easy to grow a wide range of agricultural products at a low cost when compared to other developed and emerging countries. Fresh fruit is very important for exporting, and fresh fruit detection is very important for this. It is now very easy for humans to detect rotten fruits, but it is extremely difficult for technology to detect rotten fruits. Our technology has been built in a deep subfield of machine learning to solve these challenges. As time has passed, Convolutional Networks have made great progress in the field of image processing. For identifying, classifying, and distinguishing between multiple groups of fruits, All other methodologies and algorithms fall short when compared to Deep Neural Networks. Fruit classification, on the other hand, can be challenging due to similarities in form, color, and texture between fruits in the same class. Deep learning is a sort of machine learning that has several layers and may be used to forecast and solve a wide range of issues. For solving this problem, we used a deep learning algorithm. Our project uses CNN algorithms to identify fresh and rotten fruits using five models: Xception, Inception V3, VGG16, MobileNet, and

NASNetMobile. Mango, Banana, Apple, Plum, and Orange are five common fruits in Bangladesh. There are two kinds of data in every dataset: fresh and rotten. From there, 4532 are in training, and 1126 are in the project's test dataset. For starters, we classify the fruits for a variety of reasons. In this study, we used deep learning models.

1.2 Motivation

Automatically recognizing fruits using computer vision remains a tough job due to similarities in colors and textures, as well as diverse varieties and characteristics such as their position or lighting circumstances.We want to test five well-known models the VGG16, InceptionV3, Xception, MobileNet, and NASNetMobile that use deep convolutional neural networks and apply them to detecting Bangladeshi fresh fruits. Using this methodology, which is both innovative and useful, we would be able to assess the use of these models as well as their success on our own Bangladeshi fruits. As a consequence of this effort, we will be able to distinguish between rotting and fresh fruits, which will benefit our economy. Apart from that, in order to stay well in this pandemic situation, we must consume fresh fruits, which is why it is important to consume fresh fruits. In this situation, our work would be immensely valuable.

1.3 Questions to Think

- How Does Fresh or Rotten Fruit Identification Impact Our Everyday Lives?
- Is it possible to use machine learning and deep learning models at the same time?
- What are the difficulties that Fruit Identification Applications face?
- Which method is best for detecting fresh or rotten fruit?
- What are the areas of Fruit Detection where my research interests should be pursued?

1.4 Expected Outcome

Fruit detection is a more complex sort of fine-grained computer vision than ordinary picture recognition. Both academic and industrial potential exists for auto fruit categorization. If I train datasets, we will be able to discriminate fresh fruits from decaying fruits using five models that we use in our job. We put five alternative models to the test and discovered that Inception V3 had the best accuracy (97.34 percent).

CHAPTER 2

BACKGROUND

2.1 Introduction:

Convolutional networks do not view images in the same way as humans do; instead, they see them as volumes. Convolutional networks perceive depth in addition to width and height while viewing an image. Due to the RGB encoding, the convolutional network ingests a picture as three separate depth layers placed one on top of another, referred to as channels.

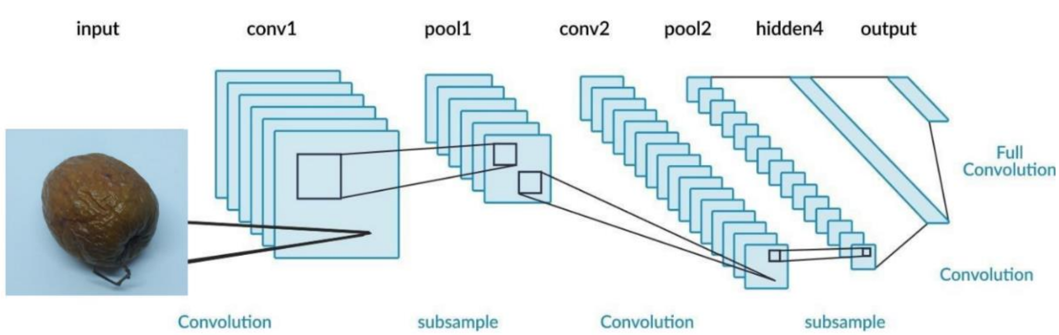

Figure 2.1: CNN Architecture.

Square patches of pixels are passed through a filter by the convolutional network. A kernel is a filter that searches for trends in pixels. The number of steps performed by the filter to traverse an underlying image is called an activation map. Pixels in images include a variety of patterns. Additional activation maps are generated as a result of these trends, resulting in a new volume. Until the pictures are larger, processing them necessitates a large amount of computational power and time. Convolutional networks tackled this concern by reducing dimensionality by methods such as filter stride and down sampling.

.

2.2 CNN Training Algorithm:

Classical neural networks are difficult to use for image recognition. For solving this dilemma and challenge, CNN is used. CNN is versatile for computer vision and can pick up design in the input image. The key benefit of CNN is that it only requires a few parameters to minimize the amount of data in the appropriate model and to learn data in a limited amount of time [16]. Layers in CNN include the following:

Convolution Layer: The convolution layer is the first layer of CNN. It functions in 32 dimensions, with two or three-dimensional input images that have weights assigned to them.

Pooling Layer: When it maintains necessary information, at each characteristic, the pooling layer reduces the quantity of data in each convolutional layer [14]. It usually sits after the activation layer. [number 16] A pooling layer can reduce the size of featured maps.

Fully Connected Input layer: This layer transforms the previous layer's output into a single vector that may be utilized as an input for the next layer [14]. In addition, it assigns a label to each image.

2.3 Related Work

Bangladesh is an agricultural country. Farmers make a living by producing flowers, fruits, and vegetables and 80% of the economy comes from this agriculture. The classification of fruits is an important task. Much more work like ours has been done to categorize these good and bad fruits based on classification.

Horea et al. [1] introduced a new work about fruit classification using Neural Network (NN) which is a dataset named Fruits-360 and downloaded from the internet. The author discussed their project why they chose to do that and could use the classifier to classify the fruits. Anuja et al. [2] used four algorithms KNN, SVM, SRC, ANN to detect fruit quality. Among them, SVM gives the best accuracy 98.48% at Fruit detection, and at the defect, accuracy is 95.72% over the database of four different fruits. Santi et al. [3] Some classifiers, such as KNN and SVM, are based on the RF algorithm are used in their paper to characterize, classify, and grade fruits using machine learning and artificial intelligence. Other classifiers are outperformed by the RF algorithm with SIFT features, which has a 96.97 percent accuracy. Hasan et al. [4] Using the deep learning method, a classification of multiplication fruits was proposed. The author attained 99 percent accuracy using TensorFlow's faster R-CNN and MobileNet methods. M. Shamim et al. [5] A framework is proposed that is based on two models: a concept for a light model with six layers of convolutional neural networks and a finetuned, pre-trained Deep learning model for visual geometry group-16. Two datasets were used by the author: the first contains clear fruit images with a 99.49 percent accuracy, and the second contains fruit ages with a 96.75 percent accuracy. Susovan et al. [6] a single feature descriptor with two different kinds of features SVM is a learning algorithm that uses features to train a dataset to identify fruit. They used SVM to construct a classification model that combined texture and color features. Overall, the accuracy rate is 83.33 percent. The key contributions of their paper are (1) a segmentation technique that works best for several colored fruit products in the natural world, and (2) an improved classification and identification method for fruits and vegetables. Kyamelia et al. [7] According to the skin defects of the fruit, determine whether it is rotten or fresh. Segmentation yielding promising results was enforced for the Enhanced

UNet UNet and an updated version of it is used (En-UNet). Under a 0.95 threshold, En-UNet has the highest mean IoU score of 0.866, while UNet has the highest score of 0.66. Their proposed model performs best in real-time rotten and fresh apple segmentation, identification, and categorization. Diclehan et al. [8] examine an image dataset containing samples of three types of fruits to decide which are new and which are rotting. Gray degree co-occurrence matrices, histograms, convolutional neural networks and plenty of other features are employed to derive characteristics of their proposed model based on framework. The classification process is entrusted to well-known classifiers based on support vector machines. Various experimental scenarios, such as binary and multi-class grouping problems, are available, for the use of features based on convolutional neural networks consistently yielding the highest success rates. Zaw et al. [9] worked on a convolutional neural network-based control system for detecting objections (CNN). Through parameter optimization, they used CNN for fruit detection and identification. For the 30 classes of 971 images, their test accuracy was close to 94 percent, indicating that the proposed system and techniques could be used for vision subsystem-based control applications. Siyuan et al. [10] Convolutional neural networks (CNN) were suggested as a method for fruit classification. A six-layer CNN was created using convolution layers, pooling layers, and completely linked layers. The results of the experiment showed that their technique performed well, it outperformed three state-of-the-art approaches: voting-based support vector machine, wavelet entropy, and genetic algorithm, with an accuracy of 91.44 percent.M Senthilarasi et al. [11] To identify the stage of ripeness of banana fruit into unripe, ripe, and overripe, a fuzzy model was developed. Their suggested research looked at the MUSA database, which included banana samples at various stages of ripening. The fuzzy model outperformed state-of-the-art algorithms in their experiments, achieving an average classification rate of 93.11 percent.

Yu-Dong et al. [12] created a convolutional neural network with 13 layers (CNN). Image rotation, Gamma correction, and noise injection were used as data augmentation techniques. Max-pooling outperforms average pooling by a slight amount, according to the researchers. With an average precision of 94.94 percent, their methodology outperformed five state-of-the-art approaches. They put their procedure to the test on photographs that were not ideal. Background fruit images have an overall precision of 89.60 percent, decay images have a score of 94.12 percent, unfocused images have a score of 91.03 percent, and occlusion images have a score of 92.55 percent. Mohammed et al. [13] based on deep learning on collab editor with python, submitted a device that distinguishes between the two grapefruit varieties, pink and white. They used a Kaggle dataset with 1,312 images, 687 pictures were used for preparation, 295 for validation, and 330 for studies, all of which belonged to two Grapefruit species.

They used deep learning convolutional neural networks to distinguish the Grapefruit type with a 100% accuracy rate. Abeer et al. [14] To classify four kinds of potatoes, researchers used a deep convolutional neural network trained on a public dataset of 2400 photos (Red, Red Washed, Sweet, and White). Their model had a test set precision of 99.5 percent, a validation accuracy of 100 percent, and a teaching accuracy of 97.45 percent. Hamdi et al. [15] An efficient machine vision system for date fruit harvesting robots has been suggested. On pre-trained models, Deep convolutional neural networks were used, along with transfer learning and fine-tuning. They created a huge picture dataset of date fruit bunches in an orchard, which they used in their research, with over 8000 images of five different date varieties at various stages of maturity. For the tasks of determining type, maturity, and harvesting decision classification, their date fruit classification models were 99.01 percent accurate, 97.25 percent, and 98.59 percent, the classification times were 20.6, 20.7, and 35.9 milliseconds respectively.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction:

A research project's methodology is the most important aspect of its work. Because of the technique, a research project will progress step by step and achieve its target flawlessly. It is also possible to learn new things with the right approach and preparation. We've also used different procedures to operate on the fruit recognition dataset. Data analysis, data preprocessing, and data augmentation are all part of the technique., Train Model (the VGG16, InceptionV3, Xception, MobileNet, and NASNetMobile models).

Work Flow for Fruit detection System:

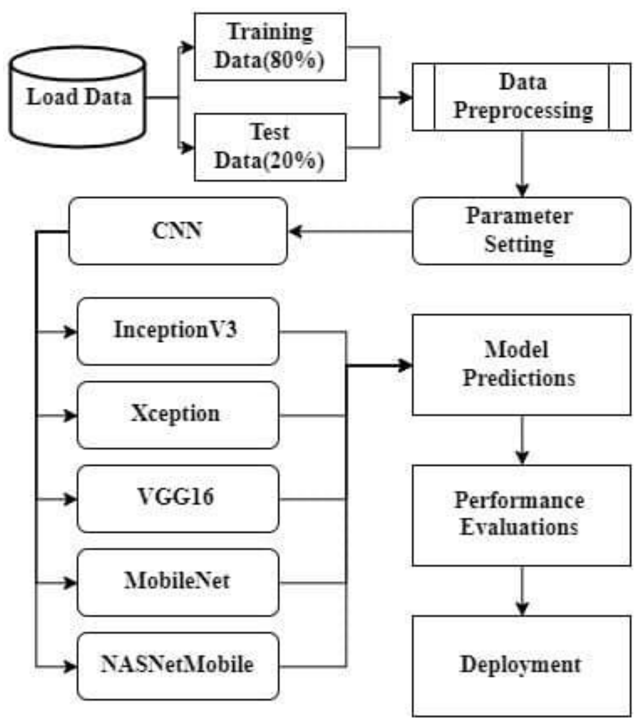

Figure 3.1: Work Flow Diagram.

3.1.1 Data Collection:

Paper data is a required component of any analysis. Accuracy would be perfect if data sets are more genuine. We gathered both fresh and rotting fruits for our project because it is about the classification of fresh and rotten fruits. Apple, peach, banana, cherry, and orange are the five varieties of new and rotting fruits we used. We used a text box to capture photographs of fruits for this project. We collected fruits from local markets and stores, then used a tent box to photograph both new and rotting fruits [17]. We've even used the internet to gather information.

3.1.2 Data Preprocessing:

Data preprocessing is the method of manipulating and converting raw data into an accessible format so that data can be more perfected. Data preprocessing is important in any research project. We started by rescaling our files, and RGB is made up of training samples. We grouped raw images into 18/20 categories in our project work. In addition, we resized our images to 224x224x3 pixels [18]. Since we use a variety of fresh and rotten fruits in the project, all photographs of fresh and rotten fruits are used in following table:

TABLE – 3.1: Image Count

Fruit name	Total Images
Apple	1160
Banana	1152
Mango	1263
Orange	1103
Plum	980
Total Images	5658

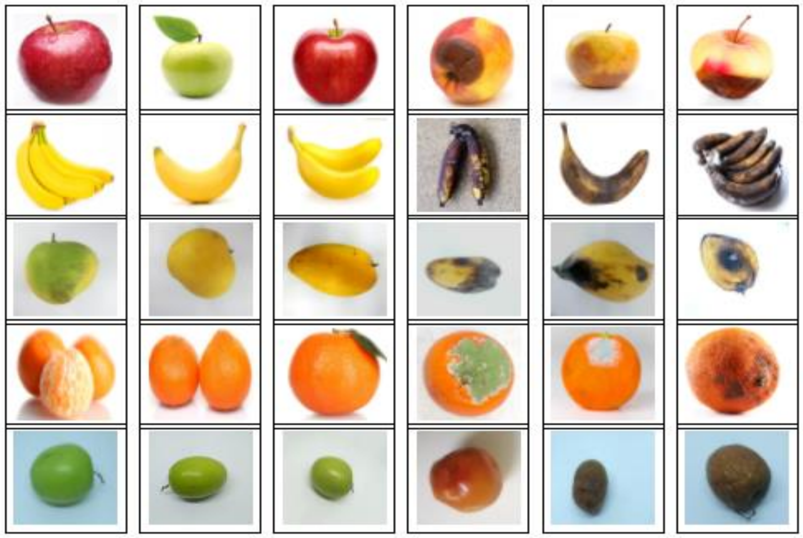

Fig 3.2: Data Sample

3.1.3 Data Augmentation:

Data augmentation is a way to enhance data by combining marginally changed data with current data. We updated datasets in our paper to achieve sufficient precision. We need to add data enlargement procedures like zooming, spinning, shearing, and transferring to our convolutional neural network model [17].

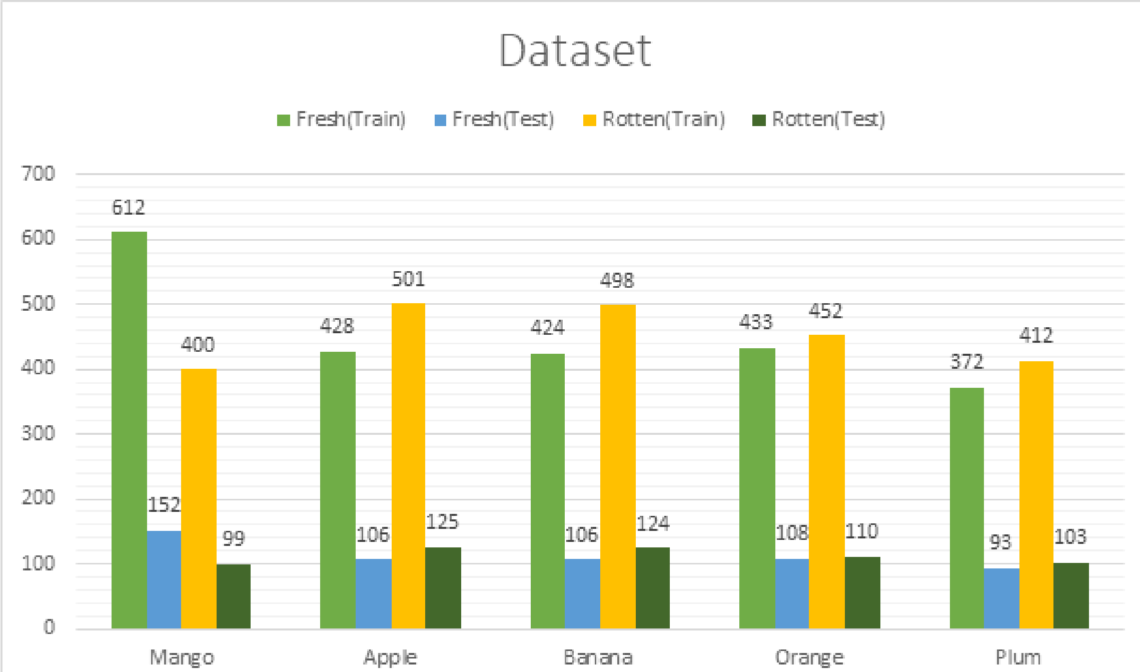

Fig. 3.3: Train & Test data Bar chart.

It represents the dataset training and test number bar chart, as seen in Fig-4. Five different kinds of fruits are obtained, with green representing fresh fruit training data, blue representing fresh test data, yellow representing rotten ride, and dark green representing rotten test break.

3.1.4 Train Model with Training Dataset:

We chose five distinct convolutional neural network models to train with our training dataset, which included 5658 photos of fruits. The following are the models we've chosen:

- INCEPTIONV3
- XCEPTION
- VGG-16
- MOBILENET
- NASNETMOBILE

3.1.5 Model Evaluation:

Its objective is to invoice a model's generalization accuracy on future data. Accuracy, precision, and recall are the three main criteria for evaluating a classification model. The fraction of correct forecasts for the test data is defined as accuracy. To find the answer, divide the number of right predictions by the total number of predictions. Our model was evaluated using the Confusion matrix, Accuracy, F1-score, Precision, Recall, and plot diagram.

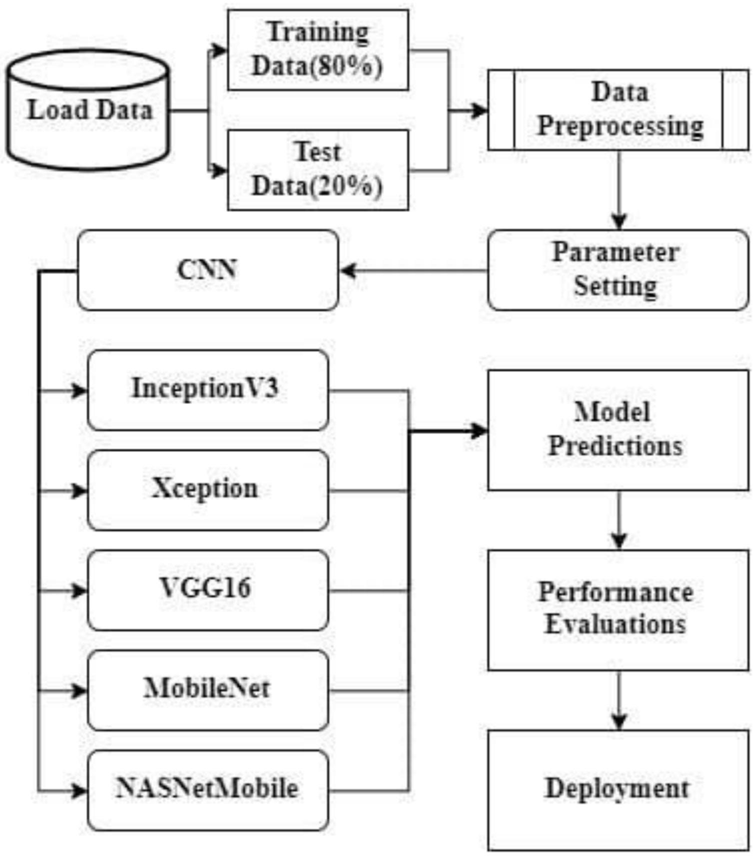

Fig. 3.4: Proposed workflow.

The InceptionV3, Xception, VGG16, MobileNet, and NASNetMobile models are summarized below:

TABLE – 3.2: Explanation of Model Architecture

Model Name	**InceptionV3**		
Layer (type)	inception_v3 (Functional)	dropout (Dropout)	dense (Dense)
Output Shape	(None, 2048)	(None, 2048)	(None, 10)
Param #	21802784	0	20490
	Total params 21,823,274	Trainable params 21,788,842	Non-trainable params 34,432
Model Name	**Xception**		
Layer (type)	xception (Functional)	dropout (Dropout)	dense (Dense)
Output Shape	(None, 2048)	(None, 2048)	(None, 10)
Param #	20861480	0	20490
	Total params 20,881,970	Trainable params 20,827,442	Non-trainable params 54,528
Model Name	**VGG16**		
Layer (type)	vgg16 (Functional)	dropout (Dropout)	dense (Dense)
Output Shape	(None, 512)	(None, 512)	(None, 10)
Param #	14714688	0	5130
	Total params 14,719,818	Trainable params 14,719,818	Non-trainable params 0
Model Name	**MobileNet**		

Layer (type)	mobilenet_1.00_224	dropout (Dropout)	dense (Dense)
Output Shape	(None, 1024)	(None, 1024)	(None, 10)
Param #	3228864	0	10250
	Total param 3,239,114	Trainable params 3,217,226	Non-trainable params 21,888
Model Name	**NASNetMobile**		
Layer (type)	NASNet (Functional)	dropout (Dropout)	dense (Dense)
Output Shape	(None, 1056)	(None, 1056)	(None, 10)
Param #	4269716	0	10570
	Total params 4,280,286	Trainable params 4,243,548	Non-trainable params 36,738

3.2 InceptionV3 Model:

It's the third installment in a series of Deep Learning Convolutional Architectures. The Tensorflow version of Inception V3 was trained using the original ImageNet dataset, which was trained with over 1 million training images, that are due to the addition of a "background" class that was not present in ImageNet's original version. Inception V3 was created for It came in second position in ImageNet's Large Visual Recognition Challenge. Transfer learning allows us to retrain an existing model's final layer, resulting in significant reductions in both training time and dataset size [19]. Inception V3 is a well-known model that can be used for transfer learning. On some extremely powerful machines, this model was trained on over a million photos from 1,000 distinct classes. Because we were able to retrain the final layer, we were able to apply the model's understanding gained during its initial training to our smaller dataset without requiring extensive preparation or processing resources. The dataset gave extraordinarily precise classifications.

Inception architecture, as opposed to VGG architecture, In terms of the amount of parameters and resource management, is more computationally efficient. Inception v3 improves on previous inception architectures by using fewer computing resources to achieve performance. Because of the possibility of reducing computational performance, making changes to the Inception network for various use cases can be difficult. For simpler paradigm adaptation, factorized convolutions, dimension reduction, regularization, and concurrent calculations are all used in the architecture.

Factorizing Convolutions:

This technique reduces a network's number of parameters while maintaining network reliability. Replacement of bigger convolutions with smaller convolutions reduces training time in the case of smaller convolutions. Consider the following scenario: Two 18-parameter 3x3 convolutions substitute a 25-parameter 5x5 convolution, resulting in fewer parameters for the algorithm to work with. When studying Asymmetric Convolutions, a 3x3 convolution is replaced by a 3x1 convolution followed by a 1x3 convolution. The 9 parameters generated by the 3x3 convolution are 6 (3x1+1x3) parameters have been replaced.

Auxiliary classifier:

During preparation, a tiny convolutional neural network is used as an auxiliary classifier between layers. They were used in Inception V1 to achieve a deeper network, However, just one auxiliary classifier is employed in Inception V3, and it functions as a regularizer.

To lower the grid scale of element maps, max pooling is typically utilized. More effective grid size reduction should be used due to disadvantages such as being too selfish or being too costly.

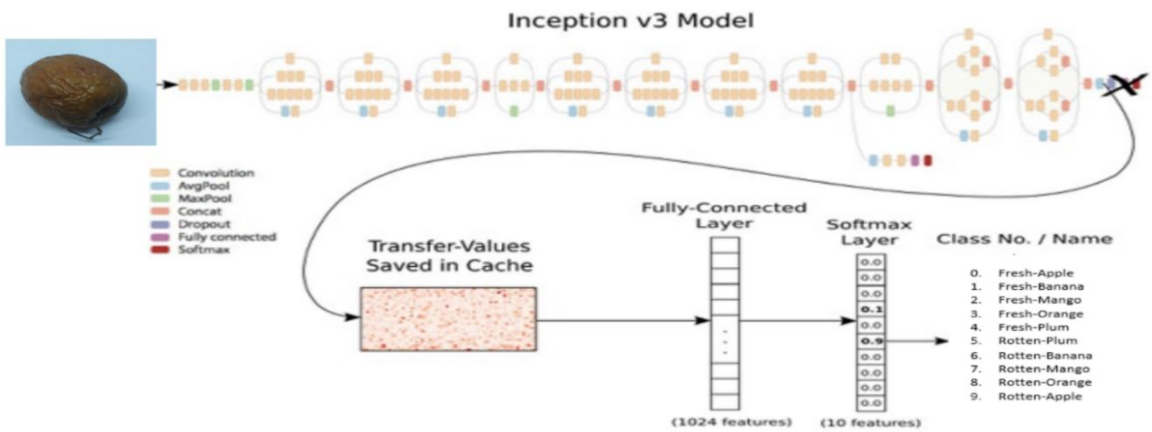

Fig. 3.5: InceptionV3 architecture

3.3 Xception Model:

When adopting a modified depth-wise separable convolution, the Xception model beats the Inception model. The Xception architecture can be compared to the Inception architecture, but it is more extreme. For instance, channel-wise nxn spatial convolution can be used to describe depthwise convolution. There would be three nxn spatial convolutions if there are three channels [20]. Pointwise convolution is a 1x1 convolution that is used to change dimensions. The original depth-wise separable convolution, on the other hand, was the polar opposite. The Xception model employs a depth-wise separable convolution that has been changed. this is a depthwise convolution after a pointwise convolutionThis transformation begins with a 1x1 convolution and progresses to nxn spatial convolutions. The model becomes much lighter and has fewer connections because convolutions across all channels are no longer required for this transition. The Xception model's overall architecture is separated into three sections: entry flow, middle flow, and exit flow. Residual connections and modified depth-wise separable convolutions are regarded the architecture's conception components exist in all three portions.

3.4 VGG-16 Model:

It is a kind of convolutional neural network. In general, the number following the network name reflects the number of layers in the architecture. The goal was to build a deep convolutional neural network that excelled at tasks by stacking layers. In the ImageNet 2014 challenge, VGG-16 won one of the prizes. After the classification vector output, choose the top five categories for assessment. ReLU serves as the activating mechanism for all secret layers. Because it speeds up learning and reduces the probability of vanishing gradient problems, ReLU is more computationally efficient.

There are 16 weight layers in the VGG-16 model. In these levels, there are 13 convolutional layers with a 3x3 filter size and three completely connected layers. The convolutional layers are divided into five categories, with a max pooling layer joining each group. The convolutional layer group's filters start at 64 and increase by a factor of two until they reach 512. The VGG-16 model's input layer takes a fixed-size RGB image of 224 × 224 pixels as well as passes it through a series of convolutional layers. It also has the option of using a 1x1 convolutional filter. The convolutional stride is set to 1 pixel when the spatial padding of this layer's input is such that the spatial resolution is preserved during convolution. The five max-pooling layers, which adopt a portion of the convolutional layers and execute max pooling with stride 2 and over a 2x2 pixel window, handle the spatial pooling.

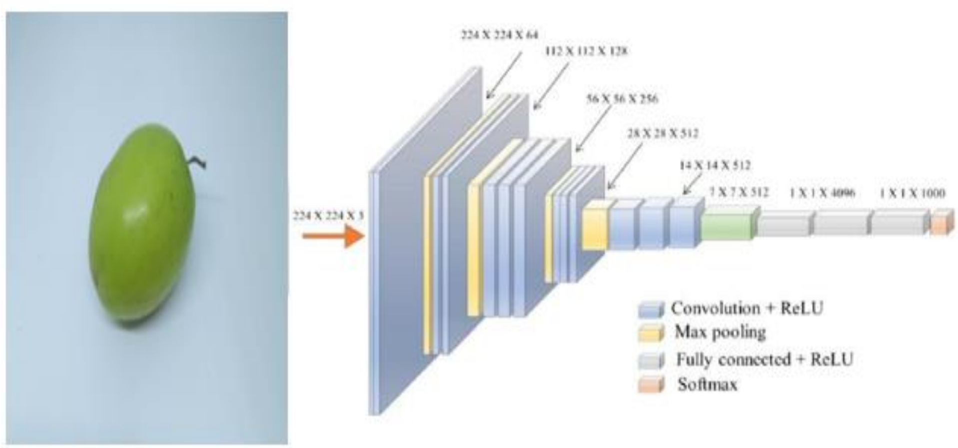

Figure 3.6: VGG-16 architecture diagram.

There are three convolutional layers, completely connected layers, with the first two having 4096 channels each and the third having seven channels, one for each class. The soft peak layer is the topmost layer. Many of the networks' entirely linked layers are set up in the same way.

3.5 MOBILENET Model:

The MobileNet model, as its name suggests, is intended for use in smartphone devices and is TensorFlow's first mobile computer vision model. The MobileNet model employs depth-wise separable convolutions, which are a type of factorized convolution, which converts a pointwise convolution into a depthwise convolution and a conventional convolution into a depthwise convolution (11 convolutions). For each input channel, MobileNets' depth wise convolution utilizes a single filter [21]. In terms of values, the results are then merged using an 11 convolution. Convolution in depth is referred to as development. In one step, a conventional convolution filters and integrates inputs to produce a new set of outputs. This is split into two layers by the depth wise separable convolution, one for filtering and the other for mixing. The number of variables is limited by factorization.

Depth wise separable convolutions are used by MobileNet. As compared to a network of normal convolutions of the same depth in the nets, it greatly decreases the number of parameters. As a consequence, lightweight deep neural networks are developed.

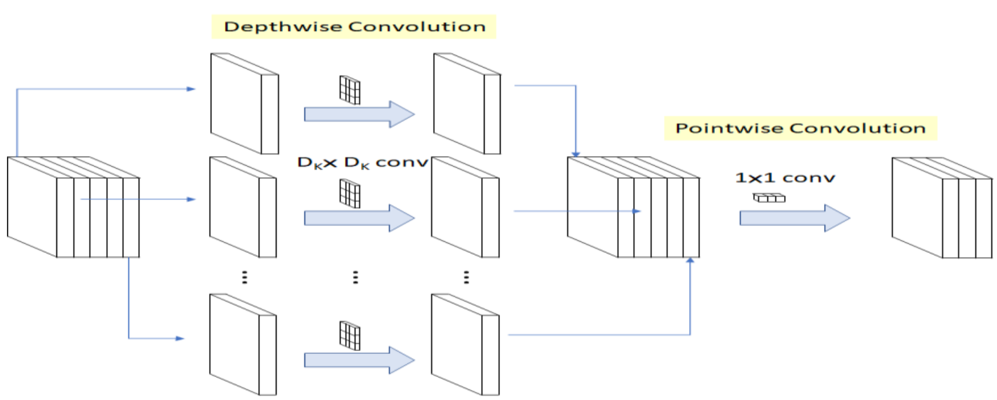

Figure 3.7: MobileNet Identity Mapping.

3.6 NASNETMOBILE Model:

It's a scalable CNN architecture composed of simple basic elements that have been optimized via reinforcement learning [22]. Convolutions and pooling are done numerous times depending on the network's required capability. In this iteration of NASNetMobile, there are 12 cells, 5.3 million parameters, and 564 million multiply-accumulates (MACs).

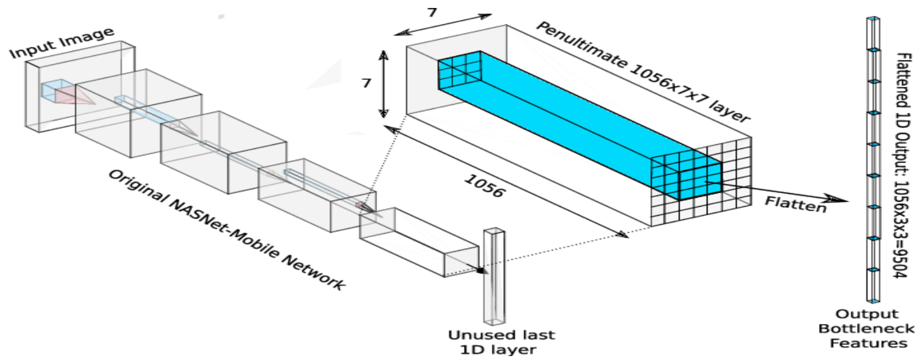

Figure 3.8: NASNetMobile Architecture

3.7 PARAMETER SETTING

We used the dataset of fresh and rotten fruits in our research. We split our dataset into two sections, with 80% of the data going into the training dataset and 20% going into the test dataset. At the same time, preparation and validation take place. We observed the effect of parameters [18] during training and tuned those parameters to produce an accurate model. Parameters for the CNN model that have been proposed are as follows: The models were educated on a set of fresh and rotting fruits, and the precision was computed on a test set.

TABLE – 3.3: Training Parameters

Batch-Size	32
Epoch	10
Training	80%
Testing	20%
Output Class	10
Input Shape	224x224x3
Train Samples	4532
Test Samples	1126

3.8 Effect of Batch Size

The batch size determines how many input samples are sent to the network. Another factor that affects classification accuracy is batch size. The larger the batch size, the longer it takes to train the dataset, and the model's precision suffers as a result, as does the memory requirement. [nineteen] As a result, when deciding on batch size, we must proceed with caution. Batch size 32 is used to run our simulations.

3.9 Effect of Number of Epochs

Epochs are nothing but the number of iterations [18]. The number of epochs is a hyperparameter that controls how many times the learning algorithm will run through the whole training dataset. Once per epoch, each sample in the training dataset had the chance to update the internal model parameters.

3.10 Effect of Optimizers

Optimizers increase the efficiency of our model by adjusting weight parameters that minimize the loss function. Our aim is to minimize the loss of our neural network by improving its parameters. The loss function calculates loss by using a neural network to match the real and expected values. [nineteen] Adam, SGD, and RMSprop were the three optimizers we used.

3.11 Confusion Matrix:

A confusion matrix is a table that shows how well a classification model performs with known true values on a set of test data. The uncertainty matrix itself is simple, but the terminology that go with it can be confusing.

(for all the classes) = (Number of Correct Classified Images)/(Number of All image)

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

Specificity=TP/ (TN+ FP)

Recall=TP/ (TP+ FN)

Precision=TP/ (TP+ FP)

Using the equations above, TP represents the number of properly classified images in a class (true positive), [23] FP is the number of photographs improperly categorized in a class (false positive), FN is the number of times an image from one class has been misidentified as an image from another. (False negative) and the number of photos that do not correspond to a class is represented by TN, and will not be labeled as such (true negative).

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Introduction:

The consistency of the proposed model is on the order of five. InceptionV3 has the highest accuracy (97.34%), precision (0.97), recall (0.97), and FI scoring (0.97) of the five models (0.97). With a 97.16 percent accuracy, precision of 0.97, recollection of 0.97, and a FI score of 0.97, the Xception model gets in second. The VGG-16 is the third-best, with 96.54 percent accuracy, 0.97 precision, .96 memory, and.97 FI score. The final two models are MobileNet and NASNetMobile. MobileNet has a 95.47 percent accuracy, 0.96 precision, 95 percent memory, and a 0.95 FI score. With a precision of 0.81, memory of 0.76, and FI score of 0.74, NaNetMobile has the lowest accuracy of 75.29 percent.

TABLE – 4.1: Final Results

Model	Accuracy	Precision	Recall	FI Score
Inception V3	97.34%	0.97	0.97	0.97
Xception	97.16%	0.97	0.97	0.97
VGG16	96.54%	0.97	0.97	0.97
MobileNet	95.47%	0.96	0.95	0.95
NasNetMobile	75.29%	0.81	0.76	0.74

4.2 InceptionV3 Model’s Experimental Results and Discussion:

Parameter:

Total params: 21,823,274

Trainable params: 21,788,842.

Non-trainable params: 21,788,842.

Accuracy:

The precision is stated for each epoch. Increasing the number of epochs enhances accuracy in this scenario.

```
Epoch 1/10
141/141 [==============================] - 1792s 12s/step - loss: 2.8699 - accuracy: 0.1807 - val_loss: 1.2482 - val_accuracy: 0.6241
Epoch 2/10
141/141 [==============================] - 44s 312ms/step - loss: 1.0504 - accuracy: 0.6417 - val_loss: 0.5552 - val_accuracy: 0.8455
Epoch 3/10
141/141 [==============================] - 44s 311ms/step - loss: 0.5671 - accuracy: 0.8268 - val_loss: 0.3282 - val_accuracy: 0.9045
Epoch 4/10
141/141 [==============================] - 44s 314ms/step - loss: 0.3503 - accuracy: 0.8987 - val_loss: 0.2435 - val_accuracy: 0.9241
Epoch 5/10
141/141 [==============================] - 44s 310ms/step - loss: 0.2573 - accuracy: 0.9264 - val_loss: 0.2025 - val_accuracy: 0.9366
Epoch 6/10
141/141 [==============================] - 44s 311ms/step - loss: 0.2064 - accuracy: 0.9425 - val_loss: 0.1774 - val_accuracy: 0.9411
Epoch 7/10
141/141 [==============================] - 44s 311ms/step - loss: 0.1669 - accuracy: 0.9548 - val_loss: 0.1603 - val_accuracy: 0.9446
Epoch 8/10
141/141 [==============================] - 44s 312ms/step - loss: 0.1415 - accuracy: 0.9625 - val_loss: 0.1438 - val_accuracy: 0.9500
Epoch 9/10
141/141 [==============================] - 44s 310ms/step - loss: 0.1216 - accuracy: 0.9666 - val_loss: 0.1332 - val_accuracy: 0.9509
Epoch 10/10
141/141 [==============================] - 44s 312ms/step - loss: 0.0864 - accuracy: 0.9786 - val_loss: 0.1251 - val_accuracy: 0.9545
```

Figure 4.1: Diagram of Accuracy for Inception V3

Confusion Matrix:

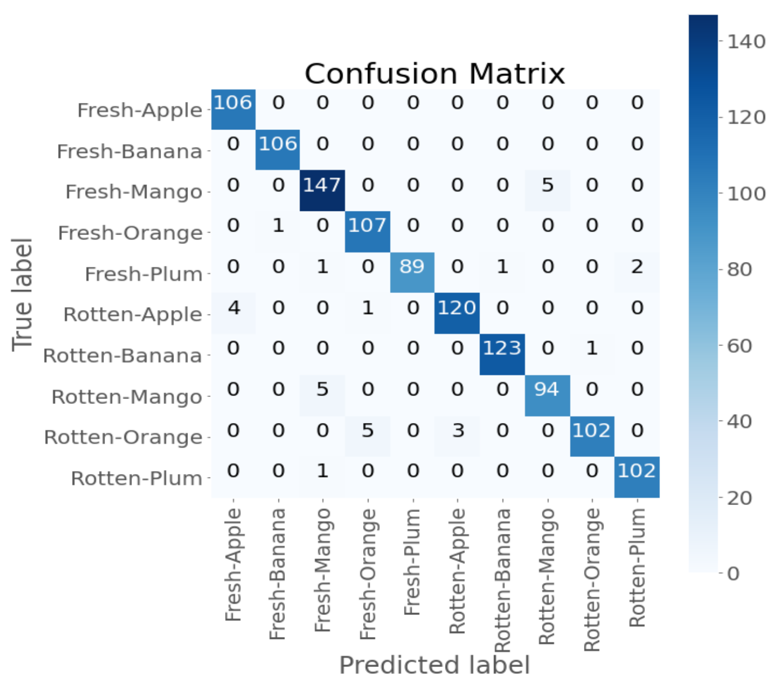

Figure 4.2: Confusion Matrix of Inception V3.

Report on Classification: Between the fruit classes, precision, recall, f1-scrore, and accuracy are shown. With this model, we were able to achieve a 97 percent accuracy rate.

	precision	recall	f1-score	support
Fresh-Apple	0.96	1.00	0.98	106
Fresh-Banana	0.99	1.00	1.00	106
Fresh-Mango	0.95	0.97	0.96	152
Fresh-Orange	0.95	0.99	0.97	108
Fresh-Plum	1.00	0.96	0.98	93
Rotten-Apple	0.98	0.96	0.97	125
Rotten-Banana	0.99	0.99	0.99	124
Rotten-Mango	0.95	0.95	0.95	99
Rotten-Orange	0.99	0.93	0.96	110
Rotten-Plum	0.98	0.99	0.99	103
accuracy			0.97	1126
macro avg	0.97	0.97	0.97	1126
weighted avg	0.97	0.97	0.97	1126

Figure 4.3: Report on Classification for Inception V3.

Training Accuracy and Validation Accuracy:

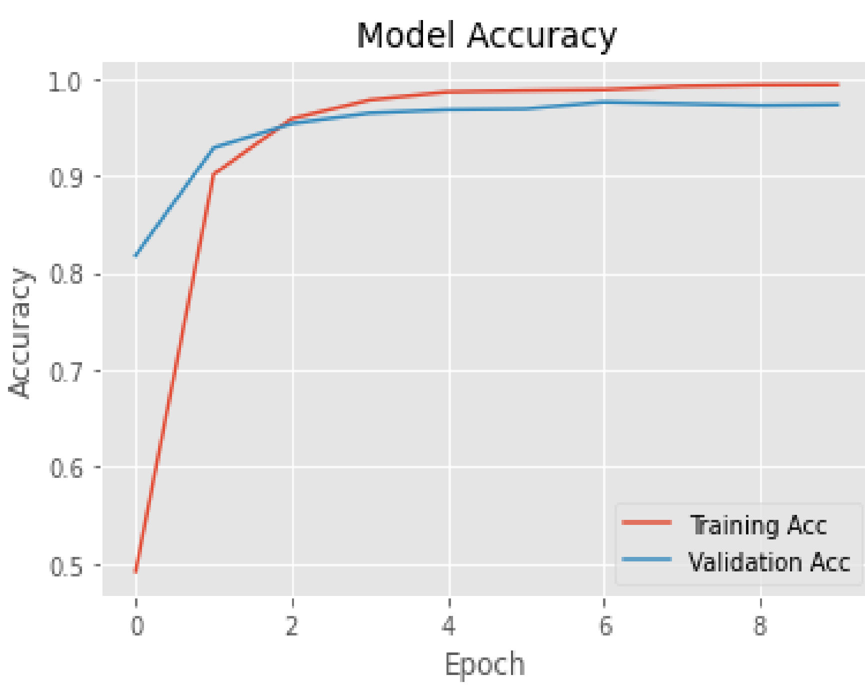

Figure 4.4: Training Accuracy and Validation Accuracy for Inception V3.

Training Loss and Validation Loss: Validation loss hit a nadir and then began to rise

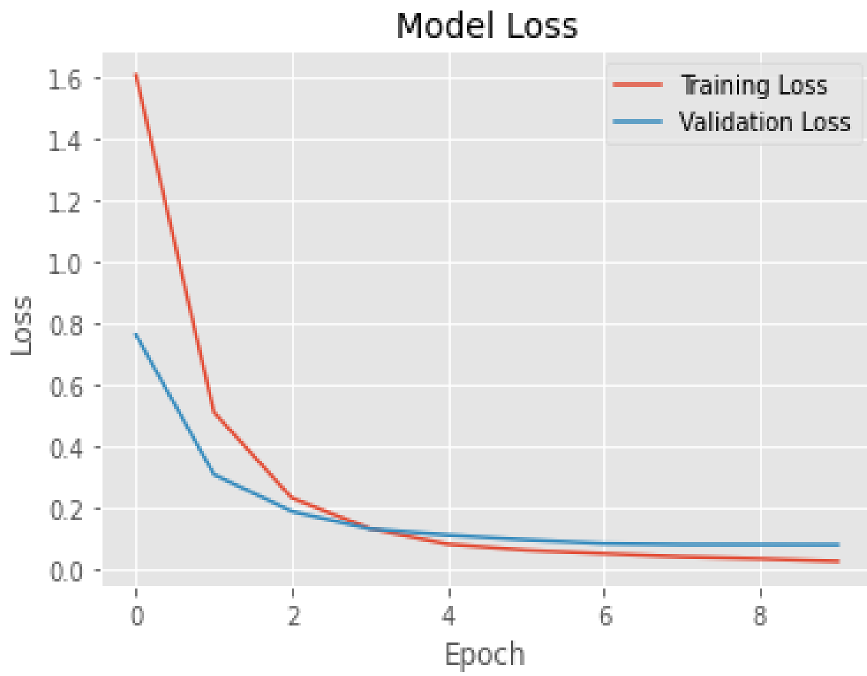

Figure 4.5: Training Loss and Validation Loss for InceptionV3

4.3 Xception Model's Experimental Results and Discussion:

Parameter:

Total paramrs: 20,881,970

Trainable params: 20,827,442

Non-trainable params: 54,528

Accuracy:

For each number of epochs, the accuracy is provided. Increasing the number of epochs causes uneven accuracy in this case.

```
Epoch 1/10
141/141 [==============================] - 2932s 21s/step - loss: 2.0702 - accuracy: 0.3028 - val_loss: 1.1433 - val_accuracy: 0.7821
Epoch 2/10
141/141 [==============================] - 188s 1s/step - loss: 0.9618 - accuracy: 0.8671 - val_loss: 0.5055 - val_accuracy: 0.9045
Epoch 3/10
141/141 [==============================] - 189s 1s/step - loss: 0.4355 - accuracy: 0.9467 - val_loss: 0.2714 - val_accuracy: 0.9554
Epoch 4/10
141/141 [==============================] - 188s 1s/step - loss: 0.2272 - accuracy: 0.9723 - val_loss: 0.1844 - val_accuracy: 0.9652
Epoch 5/10
141/141 [==============================] - 189s 1s/step - loss: 0.1360 - accuracy: 0.9823 - val_loss: 0.1404 - val_accuracy: 0.9670
Epoch 6/10
141/141 [==============================] - 188s 1s/step - loss: 0.0944 - accuracy: 0.9895 - val_loss: 0.1191 - val_accuracy: 0.9661
Epoch 7/10
141/141 [==============================] - 188s 1s/step - loss: 0.0726 - accuracy: 0.9900 - val_loss: 0.1039 - val_accuracy: 0.9741
Epoch 8/10
141/141 [==============================] - 188s 1s/step - loss: 0.0533 - accuracy: 0.9938 - val_loss: 0.1022 - val_accuracy: 0.9679
Epoch 9/10
141/141 [==============================] - 188s 1s/step - loss: 0.0497 - accuracy: 0.9916 - val_loss: 0.0871 - val_accuracy: 0.9723
Epoch 10/10
141/141 [==============================] - 189s 1s/step - loss: 0.0369 - accuracy: 0.9945 - val_loss: 0.0893 - val_accuracy: 0.9714
```

Figure 4.6: Diagram of Accuracy for Xception.

Confusion matrix:

The correctness of the identification is stated. Accurate identification of fresh and rotting fruits can be found here.

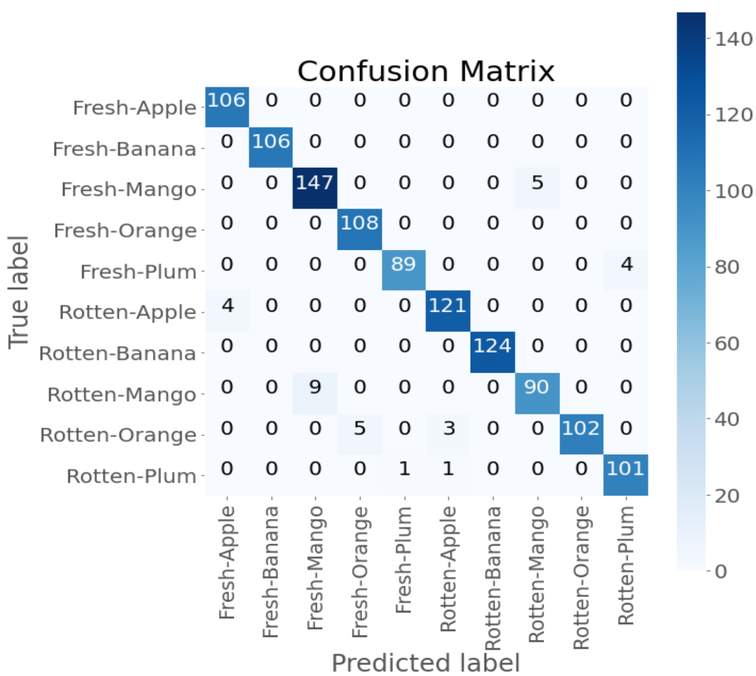

Figure 4.7: Confusion Matrix of Xception.

Report on Classification:

Between the fruit groups, precision, recall, f1-scrore, and accuracy are shown. With this model, we were able to accomplish a 97 percent accuracy rate.

	precision	recall	f1-score	support
Fresh-Apple	0.96	1.00	0.98	106
Fresh-Banana	1.00	1.00	1.00	106
Fresh-Mango	0.94	0.97	0.95	152
Fresh-Orange	0.96	1.00	0.98	108
Fresh-Plum	0.99	0.96	0.97	93
Rotten-Apple	0.97	0.97	0.97	125
Rotten-Banana	1.00	1.00	1.00	124
Rotten-Mango	0.95	0.91	0.93	99
Rotten-Orange	1.00	0.93	0.96	110
Rotten-Plum	0.96	0.98	0.97	103
accuracy			0.97	1126
macro avg	0.97	0.97	0.97	1126
weighted avg	0.97	0.97	0.97	1126

Figure 4.8: Report on Classification for Xception.

Training Accuracy and Validation Accuracy: There is a significant distinction between the curves.

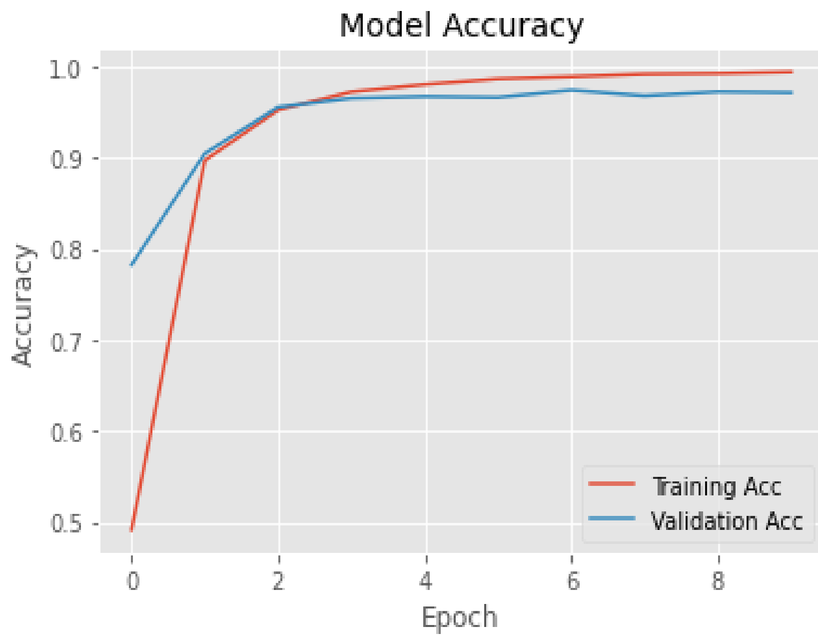

Figure 4.9: Training Accuracy and Validation Accuracy for Xception.

Training Loss and Validation Loss: The validation loss has been gradually increasing, meanwhile the train loss has been steady.

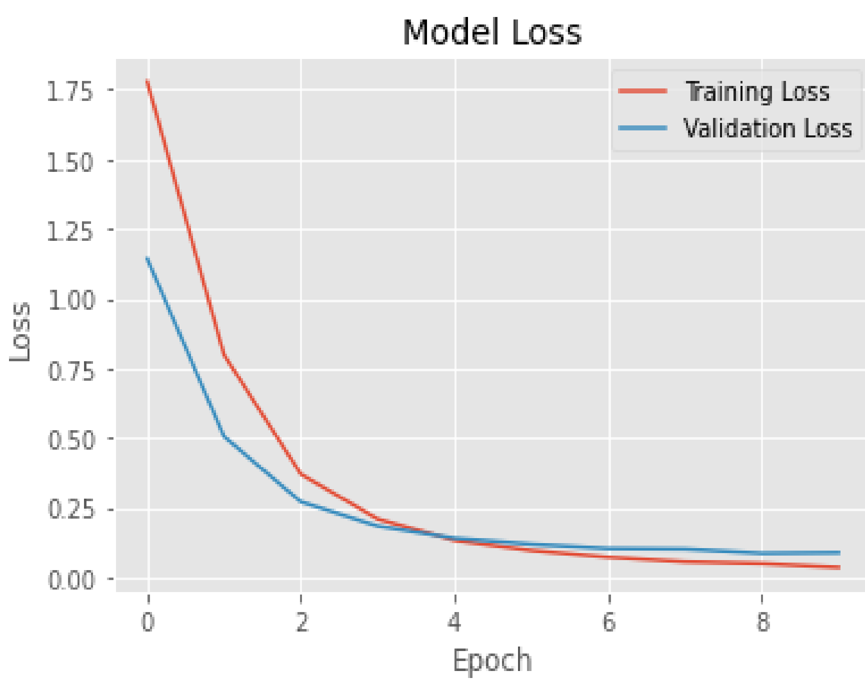

Figure 4.10: Training Loss and Validation Loss for Xception.

4.4 VGG-16 Model's Experimental Results and Discussion

Parameter:

Total params: 14,719,818.

Trainable params: 14,719,818.

Non-trainable params:

Accuracy:

The precision is reported for each number of epochs. In this scenario, increasing the number of enhances accuracy.

```
Epoch 1/10
141/141 [==============================] - 1370s 9s/step - loss: 1.9176 - accuracy: 0.3443 - val_loss: 0.3358 - val_accuracy: 0.8973
Epoch 2/10
141/141 [==============================] - 62s 442ms/step - loss: 0.4022 - accuracy: 0.8712 - val_loss: 0.2061 - val_accuracy: 0.9232
Epoch 3/10
141/141 [==============================] - 64s 450ms/step - loss: 0.2055 - accuracy: 0.9367 - val_loss: 0.2626 - val_accuracy: 0.9250
Epoch 4/10
141/141 [==============================] - 64s 456ms/step - loss: 0.1322 - accuracy: 0.9637 - val_loss: 0.1417 - val_accuracy: 0.9482
Epoch 5/10
141/141 [==============================] - 65s 461ms/step - loss: 0.0991 - accuracy: 0.9698 - val_loss: 0.1314 - val_accuracy: 0.9527
Epoch 6/10
141/141 [==============================] - 65s 463ms/step - loss: 0.0780 - accuracy: 0.9754 - val_loss: 0.1259 - val_accuracy: 0.9527
Epoch 7/10
141/141 [==============================] - 66s 464ms/step - loss: 0.0607 - accuracy: 0.9839 - val_loss: 0.0899 - val_accuracy: 0.9679
Epoch 8/10
141/141 [==============================] - 66s 465ms/step - loss: 0.0459 - accuracy: 0.9871 - val_loss: 0.1015 - val_accuracy: 0.9625
Epoch 9/10
141/141 [==============================] - 66s 465ms/step - loss: 0.0502 - accuracy: 0.9863 - val_loss: 0.0983 - val_accuracy: 0.9634
Epoch 10/10
141/141 [==============================] - 66s 463ms/step - loss: 0.0353 - accuracy: 0.9880 - val_loss: 0.1073 - val_accuracy: 0.9652
```

Figure 4.11: Diagram of Accuracy for VGG-16.

Confusion matrix:

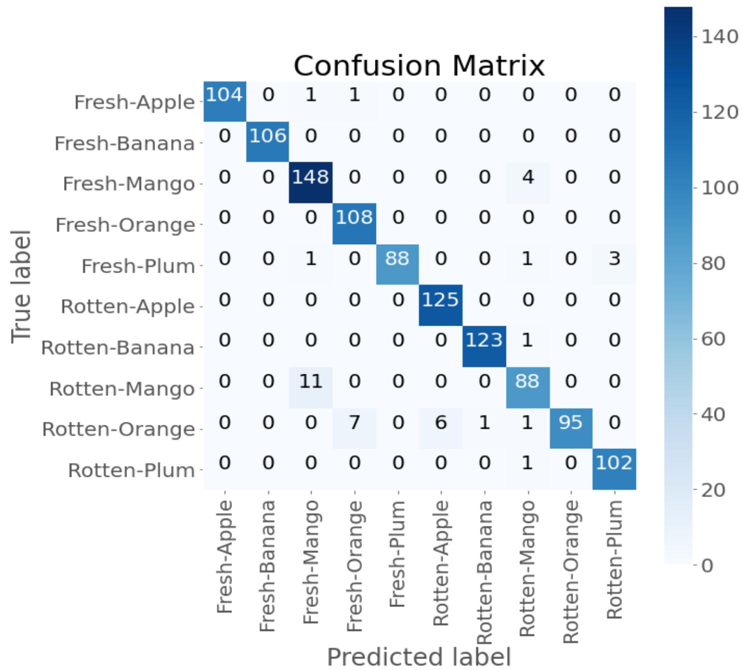

Figure 4.12: Confusion Matrix of VGG-16

Classification Report:

Between the fruit classes, precision, recall, f1-scrore, and accuracy are shown. With this model, we were able to achieve a 97 percent accuracy rate.

	precision	recall	f1-score	support
Fresh-Apple	1.00	0.98	0.99	106
Fresh-Banana	1.00	1.00	1.00	106
Fresh-Mango	0.92	0.97	0.95	152
Fresh-Orange	0.93	1.00	0.96	108
Fresh-Plum	1.00	0.95	0.97	93
Rotten-Apple	0.95	1.00	0.98	125
Rotten-Banana	0.99	0.99	0.99	124
Rotten-Mango	0.92	0.89	0.90	99
Rotten-Orange	1.00	0.86	0.93	110
Rotten-Plum	0.97	0.99	0.98	103
accuracy			0.97	1126
macro avg	0.97	0.96	0.97	1126
weighted avg	0.97	0.97	0.97	1126

Figure 4.13: Report on Classification for VGG-16

Training Accuracy and Validation Accuracy: Gradual narrowing of the gap between the curves.

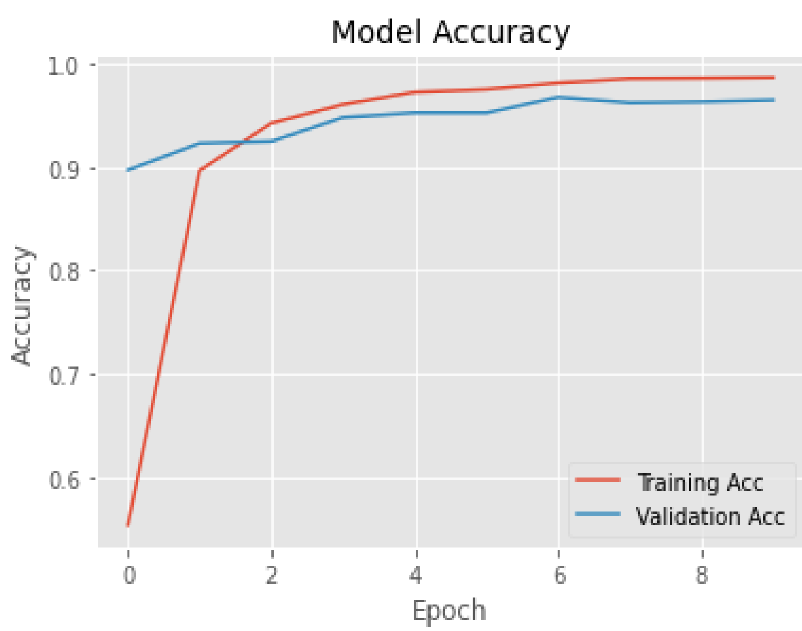

Figure 4.14: Training Accuracy and Validation Accuracy of VGG-16.

Training Loss and Validation Loss: At first, validation loss fluctuated but decreased as time went on.

Figure 4.15: Training Loss and Validation Loss for VGG-16.

4.5 MobileNet Model's Experiment Results and Discussion:

Parameter:

Total params: 3,239,114

Trainable params: 3,217,226..

Non-trainable params: 21,888

Accuracy:

For each number of epochs, the accuracy is provided. Increasing the number of epochs increases the accuracy in this case

```
Epoch 1/10
141/141 [==============================] - 1792s 12s/step - loss: 2.8699 - accuracy: 0.1807 - val_loss: 1.2482 - val_accuracy: 0.6241
Epoch 2/10
141/141 [==============================] - 44s 312ms/step - loss: 1.0504 - accuracy: 0.6417 - val_loss: 0.5552 - val_accuracy: 0.8455
Epoch 3/10
141/141 [==============================] - 44s 311ms/step - loss: 0.5671 - accuracy: 0.8268 - val_loss: 0.3282 - val_accuracy: 0.9045
Epoch 4/10
141/141 [==============================] - 44s 314ms/step - loss: 0.3503 - accuracy: 0.8987 - val_loss: 0.2435 - val_accuracy: 0.9241
Epoch 5/10
141/141 [==============================] - 44s 310ms/step - loss: 0.2573 - accuracy: 0.9264 - val_loss: 0.2025 - val_accuracy: 0.9366
Epoch 6/10
141/141 [==============================] - 44s 311ms/step - loss: 0.2064 - accuracy: 0.9425 - val_loss: 0.1774 - val_accuracy: 0.9411
Epoch 7/10
141/141 [==============================] - 44s 311ms/step - loss: 0.1669 - accuracy: 0.9548 - val_loss: 0.1603 - val_accuracy: 0.9446
Epoch 8/10
141/141 [==============================] - 44s 312ms/step - loss: 0.1415 - accuracy: 0.9625 - val_loss: 0.1438 - val_accuracy: 0.9500
Epoch 9/10
141/141 [==============================] - 44s 310ms/step - loss: 0.1216 - accuracy: 0.9666 - val_loss: 0.1332 - val_accuracy: 0.9509
Epoch 10/10
141/141 [==============================] - 44s 312ms/step - loss: 0.0864 - accuracy: 0.9786 - val_loss: 0.1251 - val_accuracy: 0.9545
```

Figure 4.16: Diagram of Accuracy for MobileNet.

Confusion Matrix:

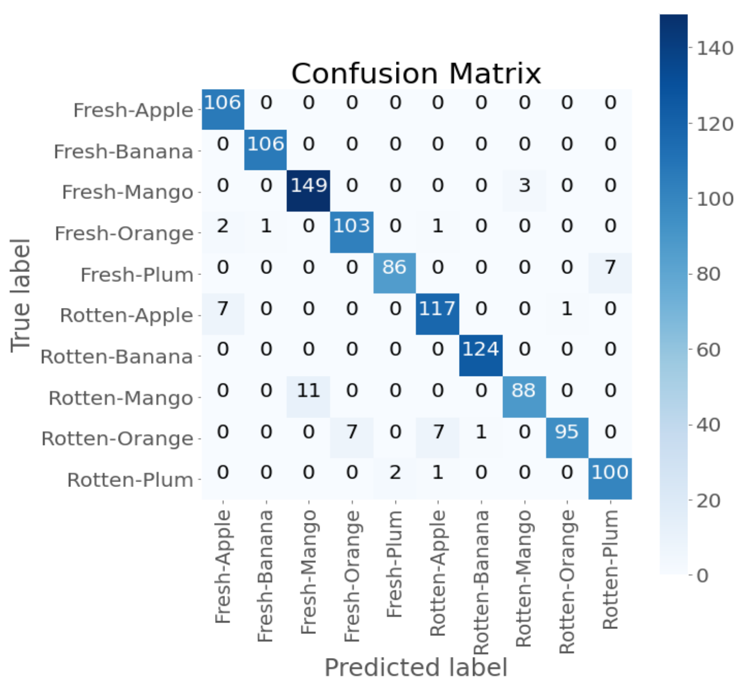

Figure 4.17: Confusion Matrix of MobileNet.

Report on Classification:

Between the fruit classes, precision, recall, f1-scrore, and accuracy are shown. With this model, was able to get a 95% accuracy rate.

	precision	recall	f1-score	support
Fresh-Apple	0.92	1.00	0.96	106
Fresh-Banana	0.99	1.00	1.00	106
Fresh-Mango	0.93	0.98	0.96	152
Fresh-Orange	0.94	0.96	0.95	107
Fresh-Plum	0.98	0.92	0.95	93
Rotten-Apple	0.93	0.94	0.93	125
Rotten-Banana	0.99	1.00	1.00	124
Rotten-Mango	0.97	0.89	0.93	99
Rotten-Orange	0.99	0.86	0.92	110
Rotten-Plum	0.93	0.97	0.95	103
accuracy			0.95	1125
macro avg	0.96	0.95	0.95	1125
weighted avg	0.96	0.95	0.95	1125

Figure 4.18: Report on Classification for MobileNet.

Training Accuracy and Validation Accuracy:

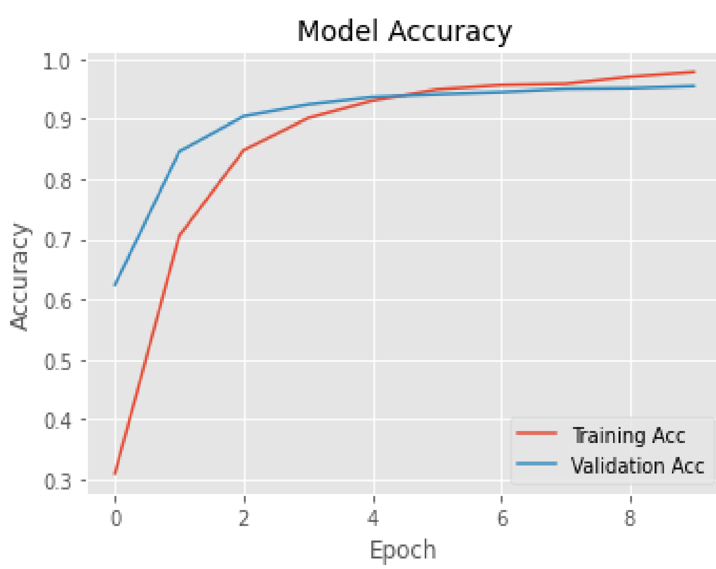

Figure 4.19: Training Accuracy and Validation Accuracy for MobileNet.

Training Loss and Validation Loss:

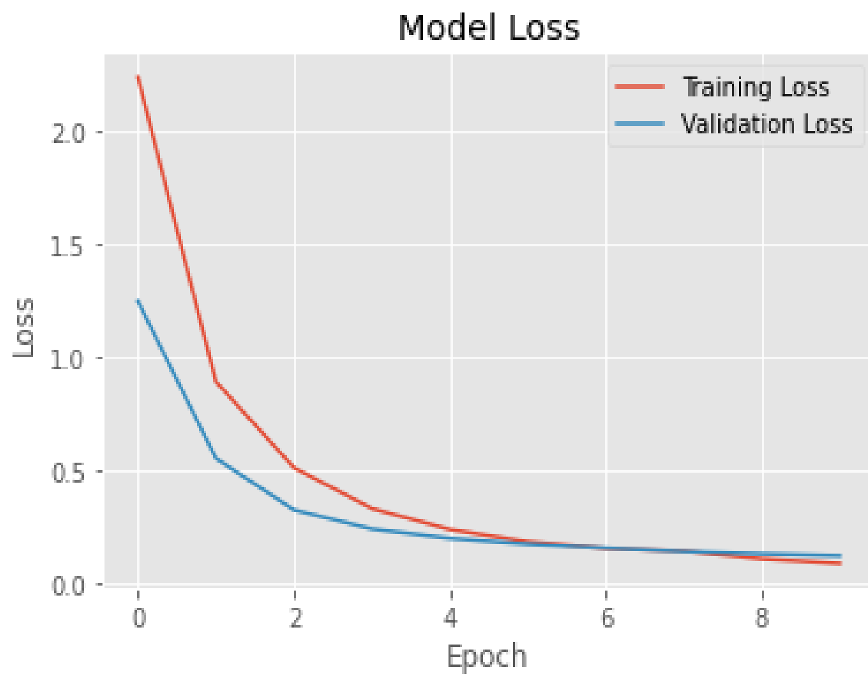

Figure 4.20: Training Loss and Validation Loss for MobileNet

4.6 NASNetMobile Model's Experimental Results and Discussion

Parameter:

Total params: 4,280,286.

Trainable params: 4,243,548.

Non-trainable param: 36,738.

Accuracy:

Each epoch has a different number of epochs, the accuracy is stated. Here we addressed train loss and train accuracy, validation loss and validation accuracy.

```
Epoch 1/10
141/141 [==============================] - 3764s 26s/step - loss: 2.3605 - accuracy: 0.1711 - val_loss: 2.0358 - val_accuracy: 0.3232
Epoch 2/10
141/141 [==============================] - 59s 417ms/step - loss: 1.4426 - accuracy: 0.5557 - val_loss: 1.7211 - val_accuracy: 0.4554
Epoch 3/10
141/141 [==============================] - 59s 415ms/step - loss: 0.8571 - accuracy: 0.7918 - val_loss: 1.4794 - val_accuracy: 0.5420
Epoch 4/10
141/141 [==============================] - 59s 415ms/step - loss: 0.5338 - accuracy: 0.8817 - val_loss: 1.3242 - val_accuracy: 0.5795
Epoch 5/10
141/141 [==============================] - 59s 415ms/step - loss: 0.3447 - accuracy: 0.9302 - val_loss: 1.1757 - val_accuracy: 0.6259
Epoch 6/10
141/141 [==============================] - 59s 417ms/step - loss: 0.2625 - accuracy: 0.9417 - val_loss: 1.0680 - val_accuracy: 0.6491
Epoch 7/10
141/141 [==============================] - 59s 416ms/step - loss: 0.1910 - accuracy: 0.9610 - val_loss: 0.9813 - val_accuracy: 0.6634
Epoch 8/10
141/141 [==============================] - 59s 416ms/step - loss: 0.1466 - accuracy: 0.9691 - val_loss: 0.8613 - val_accuracy: 0.7027
Epoch 9/10
141/141 [==============================] - 59s 416ms/step - loss: 0.1156 - accuracy: 0.9776 - val_loss: 0.7842 - val_accuracy: 0.7321
Epoch 10/10
141/141 [==============================] - 59s 415ms/step - loss: 0.0986 - accuracy: 0.9781 - val_loss: 0.7254 - val_accuracy: 0.7536
```

Figure 4.21: Diagram of Accuracy for NASNetMobile

Confusion Matrix:

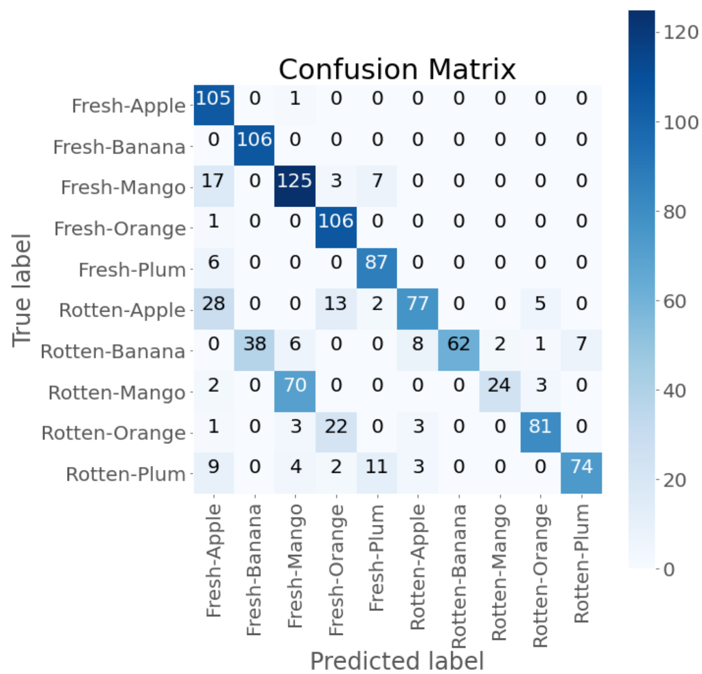

Figure 4.22: Confusion Matrix of NASNetMobile.

Report on Classification:

Between the fruit classes, precision, recall, f1-scrore, and accuracy are shown. With this model, was able to get a 75% accuracy rate.

	precision	recall	f1-score	support
Fresh-Apple	0.62	0.99	0.76	106
Fresh-Banana	0.74	1.00	0.85	106
Fresh-Mango	0.60	0.82	0.69	152
Fresh-Orange	0.73	0.99	0.84	107
Fresh-Plum	0.81	0.94	0.87	93
Rotten-Apple	0.85	0.62	0.71	125
Rotten-Banana	1.00	0.50	0.67	124
Rotten-Mango	0.92	0.24	0.38	99
Rotten-Orange	0.90	0.74	0.81	110
Rotten-Plum	0.91	0.72	0.80	103
accuracy			0.75	1125
macro avg	0.81	0.76	0.74	1125
weighted avg	0.80	0.75	0.74	1125

Figure 4.23: Report on Classification for NASNetMobile

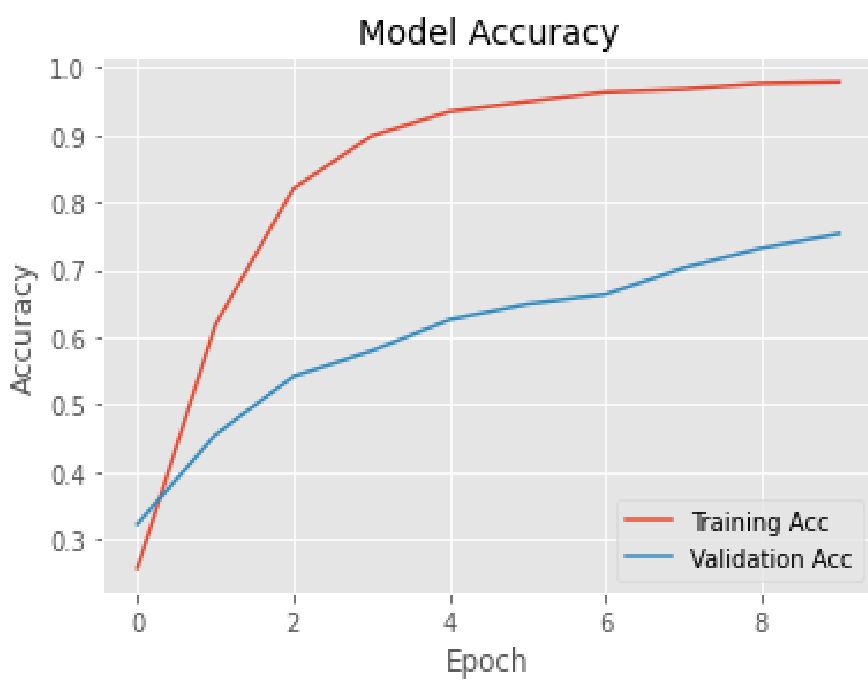

Figure 4.24: Training Accuracy and Validation Accuracy for NASNetMobile

Training Loss and Validation Loss: Noticeable fluctuations on the loss curve for validation.

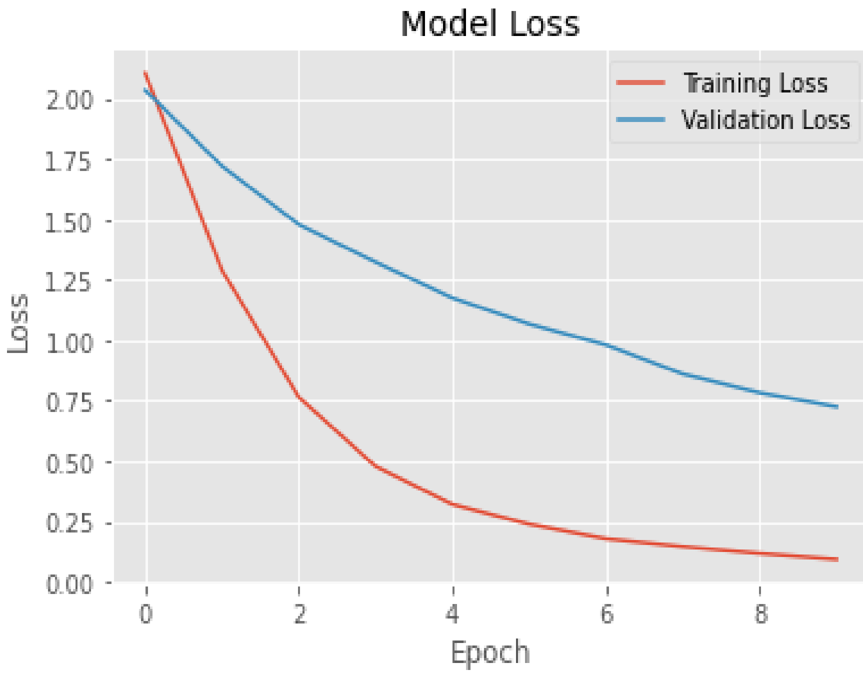

Figure 4.25: Training Loss and Validation Loss for NASNetMobile

4.7 Discussion Points:

The InceptionV3 Model's training accuracy and loss, as well as the validation accuracy and loss. In the first epoch, the training accuracy reached 29.25% and the validation accuracy was 81.79%. After 10 epochs it was 2.77% for Training loss and 8.12% for validation loss. As well as the training accuracy 99.18% and validation accuracy 97.32%. The figure of the Confusion Matrix illustrates that 30 of the 1126 images in the test set were incorrectly predicted by the InceptionV3 Model. For instance, In the case of fresh mango fruits, 147 of the 152 images predicted correctly, while only 5 images predicted incorrectly.

The training accuracy and loss of the Xception Model, as well as the validation accuracy and loss, are shown in the illustration. In the first epoch, the training accuracy was 30.28 percent and the validation accuracy was 78.21 percent. After ten epochs, the training loss was 3.69% and validation loss 8.93% as well as training accuracy 99.45%, and validation accuracy 97.14%. As shown in the figure of the Confusion Matrix, the Xception Model predicted 32 of the 1126 images in the test set incorrectly. For example, out of 99 photos of rotten mango fruits, 90 gave correct predictions and 9 gave incorrect predictions.

The VGG16 Model's training accuracy and loss, as well as the validation accuracy and loss, are depicted in the diagram. In the first epoch, the training accuracy was 34.43% and the validation accuracy was 89.73%. The training accuracy came at 98.80%, and the validation accuracy came at 96.52%, with a training loss of 3.53% and a validation loss of 10.73% after ten epochs. The VGG16 Model predicted 39 of the 1126 images in the test set incorrectly, as shown in the Confusion Matrix figure. For example, out of 99 photos of rotten mango fruits, 88 predicted correctly, while 11 predicted incorrectly.

The training accuracy and loss of the MobileNet Model, as well as the validation accuracy and loss, are shown in the diagram. In the first epoch, the training accuracy was 18.07 percent and the validation accuracy was 62.41 percent. After ten epochs, the training accuracy was 97.86%, and the validation accuracy was 95.45%, with a training loss of 8.64% and a validation loss of 12.51%. As seen in the Confusion Matrix, the MobileNet Model predicted 51 of the 1126 images in the test set incorrectly. For example, from 103 images of rotten plum fruits, 100 were correctly predicted, while 3 were incorrectly predicted.

The NASNetMobile Model's training accuracy and loss, as well as the validation accuracy and loss, are depicted in the diagram. In the first epoch, the validation accuracy was 32.32 percent, while the training accuracy was 17.11 percent. After ten epochs, the training accuracy was 97.81 percent and the validation accuracy was 75.36 percent, with a training loss of 0.986 and a validation loss of 0. 7254.The NASNetMobile Model predicted 278 of the 1126 images in the test set incorrectly, as seen in the Confusion Matrix. For instance, 74 out of 103 images of rotten plum fruits were correctly predicted, while 29 were incorrectly predicted.According to the findings, it is clear that the InceptionV3 Model performed the best, with a 97.34 percent accuracy rate.

TABLE- 4.2: Comparative Table

Authors Paper	Used Models	Dataset	Accuracy(best)
Horea et al. [1]	Convolutional Neural Network	Fruits-360, 38409 images of 60 fruits	96.3%.
Anuja et al. [2]	k-nearest neighbor (k-NN), support vector machine (SVM), sparse representation classifier (SRC), and artificial neural network are examples of classifiers (ANN)	Total-12132	Maximum accuracy-SVM-98.48% Defeated maximum accuracy-SVM-95.72%
M. Shamim et al. [5]	six CNNs layers, Visual Geometry Group (VGG)-16	Two-color image datasets, dataset1-3011, dataset2-2987	99.75%-VGG16
Susovan et al. [6]	Support Vector Machine (SVM)	240 images- 8 different classes, in every class, there are 30 image samples	83.33%
Kyamelia et al. [7]	UNet, Enhanced UNet (En-UNet	(www.kaggle.com, sriramr, fruits fresh-and-rotten-for-classification)	97.54%-En-UNet
Siyuan et al. [10]	Six-layer CNN	1800 images	91.44%
Yu-Dong et al. [12]	13-layer convolutional neural network (CNN)	3600-image dataset with 200 images for each fruit type.	94.94%

Mumenunnessa et al. [16]	Convolution Neural Network (CNN)	More than 1000 images	93%
M S Joha et al. [17]	VGG-16, RESNETV2-152, XCEPTION, DENSENET-201, INCEPTION-V3	610708 images	VGG-16 model-100%
Sai et al. [18]	CNN model-VGG 16, VGG 19MobileNet, Xception	5989 images.	97.82%

CHAPTER 5

IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

5.1 Society's Impact

The use of Fruit recognition with DCNN has a huge societal effect. This architecture is being used in crop collection robots that are autonomous. Farmers may also use this technique to estimate yields. This could result in stable crop growth and further harvest during the year. Farmers are no longer reliant on environmental constraints.

5.2 Impact on Environment

Crop classification and identification are critical for agricultural robotics. Fruit diseases can now be identified in large clusters using deep learning techniques. It aids in the healthy growth of crops and can be used in massive greenhouses, which is environmentally friendly.

5.3 Ethical Aspects

These classification methods can be used without causing damage to the environment and can even be beneficial to it. Fruit classification may result in higher crop yields with less waste, reducing the need for artificial fruit ripening.

5.3 Sustainability Plan

Mobility applications that are smart can be built to quickly identify fruits native to a particular area or to obtain essential information such as nutritional values. These applications can also be used to keep track of stock in supermarkets.

CHAPTER 6

CONCLUSION, RECOMMENSATION AND IMPLICATION FOR FUTURE RESEARCH

6.1 Conclusions:

Five powerful deep learning models were compared and evaluated in our research project, like InceptionV3, Xception, VGG-16, MobileNet as well as NASNetMobile for Fruits that are indigenous to Bangladesh are recognized. We also talked about how they function and what applications they have. Among these five models, the InceptionV3 deep learning model has demonstrated exceptional potential for fruit recognition. For testing, we created our own dataset using ten different types of fruit. We trained and tested all five models after preprocessing the data. Our goal was to see how efficient and accurate each of the five models were in classifying fresh and rotting fruit.

The scope of this project will broaden in the future to include more fruit varieties for classification, enabling any fruit farmer to use the method. In terms of classifying fresh and rotting fruits in yield so that they can get a better selling price, the proposed work is more useful for fruit-producing growers.

6.2 Recommendations:

The output of the models of deep learning that we described is dependent on a number of variables, but all five are capable of detecting fruits. We suggest the InceptionV3 model based on the results and consistency of our own dataset. On our dataset, the InceptionV3 model had the highest accuracy rate of 97.34 percent as compared to the other four models.

6.3 Implication for Future Research:

We can include more fruit groups in future studies and concentrate on the categorization of sub-classes. Larger datasets is also a possibility, as do the majority of DCNN display when there are clear accuracy rates checked involving big datasets. Other foods and vegetables can also be used for training because these models are capable of recognizing objects in any image. Various real-life obstacles, such as varying lighting conditions and clustered fruit locations, can also be included. This will entail further research into the nature of convolutional neural networks.

REFERENCES

[1] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2018.

[2] A. Bhargava and A. Bansal, "Automatic Detection and Grading of Multiple Fruits by Machine Learning," *Food Analytical Methods*, vol. 13, no. 3, pp. 751–761, 2019.

[3] S. K. Behera, A. K. Rath, A. Mahapatra, and P. K. Sethy, "Identification, classification & grading of fruits using machine learning & computer intelligence: a review," *Journal of Ambient Intelligence and Humanized Computing*, 2020.

[4] H. Basri, I. Syarif, and S. Sukaridhoto, "Faster R-CNN Implementation Method for Multi-Fruit Detection Using Tensorflow Platform," *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, 2018.

[5] M. S. Hossain, M. Al-Hammadi, and G. Muhammad, "Automatic Fruit Classification Using Deep Learning for Industrial Applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1027–1034, 2019.

[6] S. Jana, S. Basak, and R. Parekh, "Automatic fruit recognition from natural images using color and texture features," *2017 Devices for Integrated Circuit (DevIC)*, 2017.

[7] K. Roy, S. S. Chaudhuri, and S. Pramanik, "Deep learning based real-time Industrial framework for rotten and fresh fruit detection using semantic segmentation," *Microsystem Technologies*, 2020.

[8] D. Karakaya, O. Ulucan, and M. Turkan, “A Comparative Analysis on Fruit Freshness Classification,” *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2019.

[9] Z. M. Khaing, Y. Naung, and P. H. Htut, “Development of control system for fruit classification based on convolutional neural network,” *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, 2018.

[10] S. Lu, Z. Lu, S. Aok, and L. Graham, “Fruit Classification Based on Six Layer Convolutional Neural Network,” *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, 2018.

[11] S. Marimuthu and S. M. M. Roomi, “Particle Swarm Optimized Fuzzy Model for the Classification of Banana Ripeness,” *IEEE Sensors Journal*, vol. 17, no. 15, pp. 4903–4915, 2017.

[12] Y.-D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, and S.-H. Wang, “Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation,” *Multimedia Tools and Applications*, vol. 78, no. 3, pp. 3613–3632, 2017.

[13] M. M. Abu-Saqer, S. S. Abu-Naser, and M. O. Al-Shawwa, “Type of Grapefruit Classification Using Deep Learning" PhilPapers, International Journal of Academic Information Systems Research (IJAISR), ISSN: 2643-9026,2020.

[14] A. A. Elsharif, I. M. Dheir, A. S. A. Mettleq, and S. S. Abu-Naser, " Potato Classification Using Deep Learning, "International Journal of Academic Pedagogical Research (IJAPR), ISSN: 2643-9603, 2020.

[15] H. Altaheri, M. Alsulaiman, and G. Muhammad, “Date Fruit Classification for Robotic Harvesting in a Natural Environment Using Deep Learning,” *IEEE Access*, vol. 7, pp. 117115–117133, 2019.

[16] M. Keya, B. Majumdar, and M. S. Islam, “A Robust Deep Learning Segmentation and Identification Approach of Different Bangladeshi Plant Seeds Using CNN,” *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020.

[17] M. S. J. Sagor and M. Dewan, “Local Fruits Recognition Using Convolutional Neural Network,” *Apache2 Ubuntu Default Page: It works*, 18-Sep-2020.

[18] S. S. S. Palakodati, V. R. Chirra, Y. Dasari, and S. Bulla, “Fresh and Rotten Fruits Classification Using CNN and Transfer Learning,” *Revue dIntelligence Artificielle*, vol. 34, no. 5, pp. 617–622, 2020.

[19] “Inception V3 Deep Convolutional Architecture for Classifying Acute...,” *Intel*. [Online]. Available: https://software.intel.com/content/www/us/en/develop/articles/inception-v3-deep-convolutional-architecture-for-classifying-acute-myeloidlymphoblastic.html. [Accessed: 19-Mar-2021].

[20] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *arXiv.org*, 17-Apr-2017.

[22] F. Saxen, P. Werner, S. Handrich, E. Othman, L. Dinges, and A. Al-Hamadi, “Face Attribute Detection with MobileNetV2 and NasNet-Mobile,” *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019.

[23] M. Rahimzadeh and A. Attar, “A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2,” *Informatics in Medicine Unlocked*, vol. 19, p. 100360, 2020.

An Advanced Method of Identifying Fresh and Rotten Fruits Using Deep Learning

ORIGINALITY REPORT

18%	15%	9%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	dspace.daffodilvarsity.edu.bd:8080 Internet Source	11%
2	philpapers.org Internet Source	1%
3	Siyuan Lu, Zhihai Lu, Soriya Aok, Logan Graham. "Fruit Classification Based on Six Layer Convolutional Neural Network", 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018 Publication	<1%
4	ieeexplore.ieee.org Internet Source	<1%
5	Mohammad Rahimzadeh, Abolfazl Attar, Seyed Mohammad Sakhaei. "A Fully Automated Deep Learning-based Network For Detecting COVID-19 from a New And Large Lung CT Scan Dataset", Biomedical Signal Processing and Control, 2021 Publication	<1%