# Evaluating CNN and Vision Transformer Models for Mango Leaf Variety Identification

by

Zaima Labiba
19101284
Afrin A Heram
22341059
Md.Muhtasim Hossain
19101263
Sharia Alam
19201032
Binita Khan Shakal
19301145

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

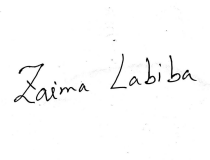
Department of Computer Science and Engineering
School of Data and Sciences
Brac University
September 2023

# Declaration

It is hereby declared that

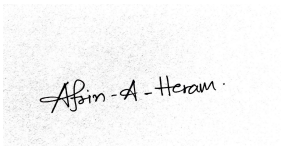
1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

**Student's Full Name & Signature:**



Zaima Labiba

19101284



Afrin A Heram

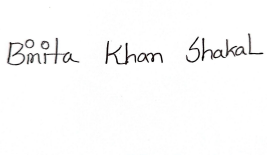
22341059

Md.Muhtasim Hossain

19101263

Sharia Alam

19201032



Binita Khan Shakal

19301145

# Approval

The thesis titled "Evaluating CNN and Vision Transformer Models for Mango Leaf Variety Identification" submitted by

1. Zaima Labiba (19101284)
2. Afrin A Heram (22341059)
3. Md.Muhtasim Hossain (19101263)
4. Sharia Alam (19201032)
5. Binita Khan Shakal (19301145)

Of Summer, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 17, 2023.

**Examining Committee:**

Supervisor:
(Member)

Dr. Amitabha Chakrabarty

Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam

Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

# Abstract

Mango, often referred to as the "King of fruits", occupies a superior place in the global agricultural landscape due to its growing demand. Thus accurate identification and classification of mango tree varieties is essential to improve quality control and inventory management in this context. In this study, we harness the power of well-established deep learning models, to detect the type and variety of mango leaves by using the mango leaf image processing method. Our meticulous analysis of accuracy and loss curves provides insight into model performance, ensuring the model is not overfitted. Additionally, we construct a comprehensive confusion matrix, highlighting the system's ability to distinguish between different mango tree varieties. We also introduced a detailed classification report, offering precision, recall, F1 score, and support for each mango tree variety. This report is a valuable tool for stakeholders, helping them make informed decisions about quality control and inventory management. Notably, we curated a vast dataset of 14,000 raw mango leaf images, collected from different locations and seasons, reflecting the diversity of mango cultivation. Our database contains 26 types of different mango leaf variants. In the proposed system, various Deep Learning and Machine Learning algorithms were utilized including VGG16, EfficientNetB3, MobileNetV2, InceptionV3, Xception, ResNet50 and ViT for classification, and a comparison was made based on their accuracy rate which is respectively 98.64%, 87.19%, 97.90%, 98.89%, 98.42%, 98.10% & 97%. By combining precision curves, loss curves, confusion matrices, and classification reports, we provide a comprehensive performance evaluation of our system. This work will bring a cathartic change in our agricultural economy by easing the process of identifying mango plants.



**Keywords:** Mango leaf classification, Identification, Mango variations, Convolutional Neural Network, Vision Transformer, Deep learning, Image classification, Image recognition, Agricultural industry, Accuracy, Pre-trained models, Dataset.

# Acknowledgement

First and foremost, we express our gratitude to Allah for enabling us to successfully accomplish our thesis within the designated timeframe and without any hindrances. In light of the aforementioned, we would like to extend our appreciation to Dr. Amitabha Chakrabarty, a highly esteemed instructor and supervisor, for his consistent support and diligent guidance, which facilitated the successful completion of our project. Furthermore, we express our gratitude to our supportive acquaintances who have provided unwavering assistance during challenging circumstances. We express our gratitude to all individuals who granted us access to their gardens, enabling us to gather data. Finally, we express our gratitude to our parents, as their steadfast support has been indispensable in our journey. With the generous support and devout prayers of others, we are now on the cusp of completing our academic journey and attaining our graduation.

# Table of Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation and Goals

Mango *(Mangifera indica)*, often hailed as the "King of fruits," holds a revered status globally, loved for its delectable taste, nutritional richness, and a myriad of varieties. In November 2010, Mango was honored as the National Tree of Bangladesh, signifying the nation's profound affection and demand for this tropical gem. Renowned for its nutrition and sweetness, mango thrives in Southeast Asian countries and graces South American landscapes as well. With over a hundred different types growing just in Bangladesh, it truly reigns supreme in the world of fruits [4]. Accurate mango leaf identification is crucial in many areas, including agriculture, horticulture, and the food business. Each variety has distinctive characteristics, including differences in tree size and leaf morphology. Mango leaf classification by kind by hand is a difficult, error-prone operation requiring specific expertise. Thus, there arises an urgent need for a deep learning system capable of automatically categorizing these varieties. Deep learning, a powerful subset of machine learning, has exhibited remarkable prowess in tackling a lot of challenges across different domains. From image classification and object detection to text generation [24] and recommendation systems, deep learning has catalyzed advancements that resonate in numerous applications [20] [21]. This burgeoning technology has substantially impacted the field of computer vision, continually pushing the boundaries of image recognition. In the agricultural sphere, mango leaf variety identification plays a pivotal role in crop management, disease detection, and yield prediction, making it a paramount area of research.

In this study, we employ popular Convolutional Neural Network (CNN) models and Vision Transformer (ViT) models for mango leaf variety identification. CNNs have long been the cornerstone of image classification, adept at automatically learning intricate features from images, capturing both low-level patterns and high-level abstractions. ViT, a recent entrant in computer vision, has showcased remarkable performance in image recognition benchmarks. This research aims to harness the capabilities of deep learning for mango species classification using leaf images. The core objective is to develop a CNN architectural model tailored for mango species classification based on leaf imagery. Crucially, we gathered a diverse and extensive dataset comprising almost 14,000 raw mango leaf images, meticulously collected from different geographical regions and across various seasons. This comprehensive

dataset mirrors the rich tapestry of mango cultivation, encompassing variations in leaf shape, color, and texture. Our deep learning models, including VGG16, InceptionV3, ResNet50, Xception, EfficientNetB3, MobileNetV2, and ViT, are fine-tuned using this dataset. Our evaluation includes the generation of accuracy curves, loss curves, confusion matrices, and classification reports. Accuracy curves offer insights into model convergence and overall performance during training. Loss curves shed light on how effectively the models reduce errors as they learn from the dataset. Confusion matrices provide a comprehensive view of the models' proficiency in classifying mango leaves into distinct categories. Classification reports offer detailed metrics, such as precision, recall, and F1 score, for each class, enabling a thorough assessment of model effectiveness. The findings of this research hold significant implications for the agricultural sector, particularly in mango tree management and disease control. Our developed classification system, driven by pre-trained CNN models, offers an efficient and reliable solution for automated mango leaf categorization. Farmers and agricultural experts can swiftly identify and categorize mango variations, facilitating timely action against disease transmission, optimization of agricultural practices, and maximization of crop yields.

This research emerges from a unique and pressing need within the agricultural landscape for the absence of comprehensive mango leaf classification research. What sets us apart is that we didn't have this dataset; we had to curate it ourselves, collecting a whopping 14,000 raw mango leaf pictures. This immense task motivated us to tackle this challenge head-on. The motivation behind this endeavor is driven by the paramount importance of addressing contemporary agricultural challenges as the increasing demand for food production, the imperative for judicious resource management and the necessity for precise crop oversight. This research seeks to fill a critical knowledge gap by pioneering the classification of mango leaf varieties. The primary goal is to harness the power of deep learning models like Convolutional Neural Networks (CNNs) and Vision Transformer (ViT), to create a robust and automated system for accurate mango leaf variety identification. This multifaceted research effort aims to not only expand the dataset but also to elevate model accuracy, enable practical deployment, explore scalability to other crops, conduct sustainability assessments, and develop user-friendly interfaces. Ultimately, our research aspires to catalyze a transformative shift in agriculture with the help of cutting-edge technology to enhance food security, optimize resource allocation, and ensure better crop quality.

## 1.2 Research Problem:

Machine learning (ML) approaches have recently gained popularity in various fields such as environmental management and agriculture. Classification of plant species based on leaves is a special research topic of this science. The mango tree is an example of a plant whose accurate identification and classification can provide important insights into farm operations, disease diagnosis, and ecosystem monitoring. Mango leaves, one of the main components of the mango tree, exhibit unique characteristics that can be used as important markers for species identification. However, mechanical identification of mango leaves based solely on visual inspection can be cumbersome and error-prone. Therefore, by integrating machine learning technology, we can greatly improve the effectiveness and accuracy of mango leaf classification.
The goal of this research project is to build a compound machine-learning model for classifying mango leaves using various parameters (color, texture, shape, etc.) and performance indicators (accuracy, recall, etc.). In this study, we collected a dataset of mango leaf photographs, pre-processed the images to extract relevant features, and then used various machine learning techniques (decision trees, random forests, neural networks, etc.) that can classify the leaves. The results of this study could help make mango leaf classification more accurate and useful, which could be used in agriculture and environmental management.
There are numerous studies related to classification of plants as classifying types of mangoes, grapes, oranges, and apples based on images processing [22]; classifying types of mangoes with CNN [18]; classifying types of mangoes, grapes, oranges, and apples based on images using CNN [7]; Classification and Grading of Harvested Mangoes Using Convolutional Neural Network [1]; All these mentioned research portrays that the identification and classification of fruits is needed. Not everyone knows about all the types of mango, and it takes experts to know the types and characteristics. Also, it can benefit the industry that works with mango fruit-based products like pickles, juice, and other snacks, etc. This study proposes a system for identifying mango plant species based on leaves using the CNN method. The main reason for preferring the CNN method from previous research is that the CNN method offers excellent accuracy till now and almost all previous studies used the leaves of the plant to identify and classify plant species. So, the main intention of our research is to propose a CNN architectural model for classifying mango species based on leaf imagery. A UNB report [32] published last year, mentioned that 200 varieties in a single tree were found in Chapainawabganj, Rajshahi. As per the report, the intention was to preserve these newly recognized species besides traditional ones in one place. So, we can say it will be a lot easier to track down the varieties even from one tree by implementing the CNN method. Also, it would further spread the reputation of the region as a mango-growing hub.
The results of this study could lead to major changes in the fields of agricultural technology and environmental management. A mixed ML model proposed to classify mango leaves can improve the accuracy of species identification while streamlining the labor-intensive identification process. The results are also useful for ecosystem monitoring, supporting sustainable agricultural practices, and early detection of diseases and insects affecting mango plants. Overall, this study aims to bridge

the gap between traditionally used manual classification methods and the growing desire for effective automated methods. Harnessing the power of machine learning models and state-of-the-art approaches can increase the accuracy and effectiveness of mango tree species identification, promoting agricultural practices and environmental protection.
When using ViT, the images of mango leaves are first transformed into sequences of fixed-size patches and then processed via a transformer-based architecture. The transformer's self-attention mechanism allows the model to capture long-range dependencies and relationships between numerous patches, as a result, the model is effective in finding distinguishing traits among various mango types. The goal can be accomplished by using ViT in the research to build a more potent feature set for precise classification and enhance the classification accuracy of recognizing mango varieties as a whole. To ascertain the benefits of utilizing ViT in this particular application, the research may compare the performance of ViT-based models with the current CNN and other machine learning models based on various performance parameters.

## 1.3 Research objective

Identifying and classifying with the bare eye which plant variety of a certain mango is very tough, challenging, and sometimes not accurate. Only the experts who work in the agriculture field can do this job and even then there can be a lack of assurance sometimes. To solve this problem, using artificial intelligence will improve the automated classification and provide accuracy. Here, we used the CNN method in the classification of mango varieties in Bangladesh. Necessary image processing methods were applied to the collected images of mango plant leaves to ensure identification was more precise. These methods include the enhancement of the images to fit for analysis and the extraction of some morphological features from binary images such as minor axis length, area, and major axis length of the leaf. We collected and processed a large dataset of mango leaf photographs to provide a diverse representation of many mango tree species and capture the features required for classification. To create a powerful feature set for accurate classification, examine and extract the following relevant aspects from mango leaf photos color, texture, and shape. Leveraging each strategy to improve classification accuracy by implementing and training mixed machine learning models using different methods such as decision trees, random forests, and neural networks. We used feature selection, parameter optimization, and model ensemble techniques to tune combined ML models, aiming to achieve the best performance in terms of accuracy, accuracy, retrieval, and computational efficiency. To determine the accuracy, efficiency, and versatility advantages of our model, we compare it to individual machine learning models and other methods for classifying mango leaves based on various performance criteria. Also, tested the resulting model against a new validation dataset and evaluated its performance in different environmental settings and image quality variations to see if it is applicable to real-world situations. We provided suggestions and insights for future research on mango leaf taxonomy, considering the limitations and difficulties of the study.

The usage of Vision Transformers (ViT) is one such strategy that can be investigated to meet the goal of the study. With its excellent performance in image identification tasks, ViT is a recent development in computer vision. The model may learn to successfully extract global context information from the images of mango leaves by introducing ViT into the classification pipeline, which may increase the accuracy of differentiating between different mango kinds.

The capacity of ViT to scale to the huge dataset is advantageous when dealing with various representations of mango tree species. To get the optimum performance in terms of accuracy and computational efficiency, ViT can also be utilized in conjunction with the current CNN-based models and other machine-learning techniques.

Our main aim is to point out that mangoes are difficult to manually classify and not everyone can visually identify them. Additionally, you can analyze your models, learn about current research, and create models that are more accurate and efficient than those currently in use. Additionally, the accuracy of the model should be evaluated and improved, and suggestions for improvement should be made. By achieving these goals, this research aims to contribute to the fields of agricultural technology and environmental management by providing a useful automatic classification method for mango leaves. The results of this study may improve accurate species identification, disease detection, and ecosystem monitoring, ultimately enhancing conservation efforts and supporting sustainable agricultural practices.

The project ultimately intends to make a contribution to the fields of agricultural technology and environmental management by offering a more precise and effective automatic classification method for mango leaves by utilizing ViT or comparable cutting-edge methodologies. The findings of this study may improve species identification, disease detection, and ecosystem monitoring, promoting sustainable agricultural practices and bolstering conservation efforts in relation to mango agriculture.

# Chapter 2

# Related Work

Researchers used ML classifiers to distinguish between different mango cultivars in Classification of *(Mangifera indica)* [30]. He following steps were included in this methodology: picture acquisition, pre-processing, segmentation, feature extraction, classification, and evaluation. Two of the classifiers, the LMT and KNN, offered decent classification accuracy to identify among the eight mango species. LMT required classification accuracy rates between 80.33% and 88.33% for a number of mango varieties, while the KNN classifier reached between 88.33% to 97%, the greatest overall categorization accuracy.

In this article [8], artificial intelligence is used to identify mango leaves. This work proposed an automatic approach for classifying mango plant leaves based on shape features and RST-invariant characteristics of the leaves. Leaf photos are processed using skillful edge detection and morphological feature extraction methods. It is believed that categorization is required in order to differentiate mango plant leaves from one another using the data gained from characteristic selection. A separate feature from the target image is compared to a feature from the cataloged image data. Images of leaves are processed using morphological feature extraction and canny edge detection algorithms. The output of the classification model is produced by an artificial neural network method based on CVIPTools. The successful method of detection is applicable to a wide range of mango species, such as mango trees gadung and curut. Up to 77.78% of detection accuracy can be achieved using the system. This result could be regarded as perfect. Two classification methods, Support Vector Machine (SVM) and Fuzzy K-Nearest Neighbor (FK-NNC), are evaluated for system enhancement in each class. The application's performance results in an accuracy of up to 88.89%. Comparing this performance to the prior one, it is noticeably better. The K-Nearest Neighbor (K-NN) algorithm experiments also produced 90% accuracy in the choice of the optimal features for mango tree species detection.

According to the study conducted by [3], the utilization of texture as the major characteristic for detection is justified by its ability to discern the physical attributes of mango leaves based on their texture. The alteration in hue is easily discernible without the aid of optical instruments.

Then, according to this study [11], Energy (C), uniformity (S), one moment (M), five moments (M), four moments (M), and a second moment are the most effective. Features of mango leaves image texture derived from Hue components (M). However, the factors cannot deliver optimal performance. The K-NN method can achieve an accuracy of 0.83 (83%). Image texture characteristics of mango leaves can provide

the utmost level of precision. The K-NN technique can achieve an accuracy of 0.89 (89%).
In this work [2], software capable of identifying a mango variety from an image of its leaf was effectively developed. Using the Otsu thresholding technique, the region of interest was isolated from the background. Nine color features, seven moments invariant features, and nine textural features were extracted using image processing techniques. Using ten leaf image samples for each of the four mango varieties, the software was effectively trained using the backpropagation method. The identification accuracy of the software is 96%. It is possible to improve the accuracy of the mango variety recognizer by utilizing more leaf image samples during training and by ensuring that the various leaf growth stages are accurately represented.
In [27], fruit classification is proposed. If expert grading is assumed to be 100% accurate, the performance accuracy of the proposed system for grading is nearly 90%. Nonetheless, this variation is a result of the subjective perception of the mango by experts, which is self-evident. In addition, the repeatability of the proposed system is determined to be 100 percent.
Using image processing and machine learning techniques, the authors of this paper [18] developed multi-parameter-based mango grading. Using image processing techniques, color, geometric, and shape-related features were extracted. These features were then used by pre-trained random forest classifiers to determine mango ripeness (unripe/middle-ripe/ripe), size (small/medium/large), and shape (well-formed/deformed). For defect segmentation, K-means clustering was used to classify mango defects as (non-defective/middle-defective/completely defective). Using a grading formula that integrates parameter-specific quality scores according to predicted categories, the final grade was determined. The classification accuracy for ripeness, size, and shape of Dashehari mangoes in a created dataset was 100%, 98.19%, and 99.20% respectively. Mangoes could be graded with an accuracy of 88.88% using formula-integrated grading.
This paper by Thakur, Khanna, Sheorey, & Ojha (2021) proposes the transformer-based programmed infection discovery show "PlantViT", which could be a cross breed of a Convolutional Neural Organize and a Vision Transformer[29]. The objective is to identify plant diseases from leaf pictures employing a profound learning method based on Vision Transformer. The show consolidates CNN and Vision Transformer capabilities. The Vision Transformer is established on a multi-head center module. The try was assessed on PlantVillage and Embrapa, two large-scale open-source datasets for plant illness location. The proposed show accomplishes 98.61% and 87.87% accuracy on the PlantVillage and Embrapa datasets, individually, agreeing to test comes about. The PlantViT can achieve critical advancement over the current state-of-the-art strategies in plant infection detection.
In this paper, the authors propose a novel methodology [33] that presents a unified approach for integrating multiple features and classifiers. This approach offers several advantages over a simplistic method where all features are concatenated and independently provided to each classification algorithm. It requires less training and is better suited for addressing specific problem domains. In addition to this, the technique that has been presented is suitable for continuous learning. This includes the process of refining a learned model as well as incorporating new classes to be distinguished. The efficacy of the proposed fusion methodology is confirmed through its application in a semi-controlled setting, specifically in the context of a

multi-class fruit-and-vegetable categorization task. This task is typically carried out in environments such as distribution centers or supermarket cashiers. The findings indicate that the proposed solution has the capability to decrease the classification error by as much as 15% points compared to the baseline.
In this inquiry about work by Supekar  Wakode (2020) a mango reviewing framework based on outside parameters specifically readiness, measure, shape, and abandons was created[19]. Picture-preparing methods were connected to extricate the color, geometric, and shape-related highlights. These highlights were assisted by pretrained irregular timberland classifiers to determine the mango readiness (unripe/mid-ripe/ripe), estimate (small/medium/large), and shape (well-formed/deformed) category. K-means clustering was connected for defect division to decide the mango imperfection category as (non-defective/mid-defective/completely-defective). Additionally, the last review was performed employing an evaluating formula that combines the parameter-specific quality scores relegated, concurring to anticipated categories. Readiness, measure, and shape classification performed on a made dataset of Dashehari mangoes accomplished a test exactness of 100%, 98.1%, and 99.20D44 separately. Formula-based coordinates evaluation might review mangoes with 88.88 % accuracy.
(Rocha et al., 2010) presents a bound-together approach that can combine numerous highlights and classifiers that require less preparation and is more satisfactory for a few issues than a gullible strategy, where all highlights are essentially concatenated and encouraged freely to each classification calculation. Other than that, the displayed strategy is amiable to nonstop learning, both when refining a learned demonstration and when including modern classes to be segregated against. The presented combination approach is approved employing a multi-class fruit-and-vegetable categorization errand in a semi-controlled environment, such as a dissemination center or a general store cashier. The comes about appears that the arrangement is able to decrease the classification mistake by up to 15 % with regard to the baseline[1].
The researchers in this study [4] utilized a substantial, complex convolutional neural network to categorize the 1.2 million high-resolution images in the ImageNet LSVRC-2010 competition into 1000 distinct classes. In the evaluation of the test data, we obtained top-1 and top-5 error rates of 37.5% and 17.0% respectively, demonstrating a significant improvement compared to the previous state-of-the-art performance. The neural network comprises five convolutional layers, some of which are accompanied by max-pooling layers, and three fully-connected layers. It is equipped with a final 1000-way softmax and possesses 60 million parameters and 650,000 neurons. In order to expedite the training process, we employed non-saturating neurons and leveraged a highly efficient GPU implementation of the convolution operation. In order to mitigate the issue of overfitting in the fully connected layers, the researchers implemented a regularization technique known as "dropout," which has been recently developed and demonstrated significant efficacy. Additionally, we submitted a modified version of this model in the ILSVRC-2012 competition and attained a first-place ranking with a top-5 test error rate of 15.3%. This outperformed the second-best submission, which achieved a test error rate of 26.2%.
In this study, the authors [6] examine the impact of the depth of a convolutional network on its accuracy in the context of large-scale image recognition. The primary contribution of our study is a comprehensive assessment of networks with progressively increasing depth, employing an architecture that incorporates compact (3×3)

convolution filters. Our findings demonstrate that by increasing the depth to 16-19 weight layers, a substantial enhancement over previous configurations can be attained. The aforementioned results served as the foundation for our participation in the ImageNet Challenge 2014. In this competition, our team achieved first and second-place rankings in the localization and classification tracks, respectively. Additionally, we demonstrate the ability of our representations to effectively generalize to alternative datasets, resulting in the attainment of state-of-the-art outcomes. The two ConvNet models that have demonstrated the highest performance have been made accessible to the public in order to encourage further investigation into the application of deep visual representations in the field of computer vision. The first-place entry demonstrated a test error rate of 15.3%, which outperformed the second-place entry's rate of 26.2%.

The algorithm proposed by Aakif and Khan (2015) consists of three distinct stages for plant identification: i) pre-processing, ii) feature extraction, and iii) classification. Various leaf features, including morphological features, Fourier descriptors, and recently proposed shape-defining features, are evaluated in comparison to the achievement of 26.2% obtained by the second-best entry[7]. The aforementioned features are utilized as the input vector for the artificial neural network (ANN). The algorithm has been trained using a dataset consisting of 817 samples of leaves obtained from 14 distinct species of fruit trees. The algorithm demonstrates a high level of accuracy, surpassing 96%. In order to assess the efficacy of the algorithm, it was subjected to testing using the Flavia and ICL datasets. The results indicate a 96% accuracy rate for both datasets.

The study by He et al. (2016) [9] introduces a residual learning framework that aims to facilitate the training of deep networks, surpassing the depth of previously employed networks. The layers are reformulated in a manner that explicitly learns residual functions by referencing the layer inputs, as opposed to learning functions without any reference. The authors present a comprehensive collection of empirical evidence that demonstrates the ease of optimization and improved accuracy achieved by residual networks when their depth is significantly increased. The authors of the study conducted an evaluation of the dataset, comparing the performance of residual networks with a depth of up to 152 layers to that of VGG networks. It was found that the Residual Networks (ResNet) were 8 times deeper than the VGG networks, while still maintaining lower complexity. A collection of these residual networks achieves an error rate of 3.57% on the ImageNet test dataset. The outcome achieved first place in the ILSVRC 2015 classification task. In addition, the authors provide an analysis of CIFAR-10 utilizing models comprising 100 and 1000 layers. The significance of the depth of representations holds great importance in various visual recognition tasks. The 28% relative improvement on the COCO object detection dataset was solely attributed to the highly profound representations employed by the researchers. The utilization of deep residual networks served as the fundamental framework for our entries in the ILSVRC and COCO 2015 competitions. Notably, our submissions achieved first place rankings in the ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation tasks.

This paper [10] provides a thorough examination of the existing body of literature with the objective of determining the current advancements in the application of convolutional neural networks (CNNs) for the purpose of diagnosing and identifying plant pests and diseases. Furthermore, this study highlights several challenges

that currently hinder the performance of the models. Additionally, it identifies areas where further research is needed to bridge existing gaps. In this context, we conduct a comprehensive examination of studies employing diverse methodologies that have focused on the detection of plant diseases, the characteristics of the datasets used, as well as the specific crops and pathogens investigated. Additionally, this paper examines the widely utilized five-step methodology for the identification of plant diseases. This methodology encompasses the stages of data acquisition, pre-processing, segmentation, feature extraction, and classification. The paper examines several deep learning architecture-based approaches that demonstrate an accelerated convergence rate in the field of plant disease recognition. This review provides insights into the emerging trends in utilizing CNN algorithms for diagnosing plant diseases. It also highlights the areas that require further attention from the research community.
The proposed system [5] is a computer vision-based approach for grading mango *(Mangifera indica)* fruits. It involves the extraction of various features that are responsive to the maturity level, size, and surface defects of the fruits. The maturity prediction task has utilized the Recursive Feature Elimination (RFE) technique in combination with Support Vector Machine (SVM) based classifiers. The determination of size and surface defects is accomplished through the utilization of various image processing methods. The system employed Multi-Attribute Decision Making (MADM) theory as a solution to address the issue of multiple characteristics. The findings indicate that the rate of size detection error is approximately 3%, while the accuracy of maturity prediction is 96% and the accuracy of surface defect detection is 92%. The grading accuracy of the proposed system is approximately 90% when assuming that expert grading is completely accurate. However, the observed variation in perceiving the mango visually is attributed to the subjective judgment of expert individuals, a fact that is readily apparent. Additionally, it has been determined that the proposed system exhibits a repeatability rate of 100.
The interpretation of Inception modules in convolutional neural networks as an intermediate step between regular convolution and the depth-wise separable convolution operation (consisting of a depth-wise convolution followed by a point-wise convolution) is presented by the authors [13] From this perspective, a depth wise separable convolution can be conceptualized as an Inception module consisting of a significantly increased number of towers. The aforementioned observation prompts us to propose a new architectural design for deep convolutional neural networks, drawing inspiration from the Inception framework. In this proposed design, the Inception modules are substituted with depth-wise separable convolutions. In this study, we demonstrate that the architecture known as Xception exhibits a slight improvement in performance compared to InceptionV3 when evaluated on the ImageNet dataset, which was specifically designed for Inception V3. Moreover, Xception demonstrates a significant performance advantage over Inception V3 when assessed on a larger image classification dataset consisting of 350 million images and 17,000 classes. Given that the Xception architecture possesses an equivalent number of parameters as the Inception V3 architecture, the observed improvements in performance can be attributed to more effective utilization of model parameters, rather than an increase in capacity.
Here, the study [31] introduces a novel disease detection model called "PlantViT," which combines a Convolutional Neural Network (CNN) with a Vision Transformer. This hybrid model is described in detail in the (Boukabouya et al., 2022). The ob-

jective of this study is to utilize a Vision Transformer-based deep learning approach to accurately detect and classify plant diseases using images of leaves. The model leverages the capabilities of Convolutional Neural Network (CNN) and the Vision Transformer. The Vision Transformer utilizes a multi-head attention module as its foundation. The experiment was assessed using two extensive open-source datasets for plant disease detection, namely PlantVillage and Embrapa. The experimental findings demonstrate that the model proposed in this study attains an accuracy of 98.61% and 87.87% on the PlantVillage and Embrapa datasets, respectively. The PlantViT demonstrates substantial advancements compared to existing state-of-the-art techniques in the field of plant disease detection. The topic of interest is plant disease. The vision transformer and convolutional neural network are both widely used models in the field of computer vision. The vision transformer is a recent architecture that has gained attention for its ability to capture long-range dependencies in images through PlantVillage an online platform that serves as a valuable resource for individuals.

The study by [16] introduced a methodology that utilizes a fuzzy logic algorithm and K-NN as a classification technique for mango leaves. The outcomes of the fuzzy logic algorithm demonstrate an accuracy rate of 80% in identifying the "Indian" mango, 72.73% for the "carabao" mango, and 80% for the "saperada" mango. The findings from the K-NN classifier indicate that when k values of 1 and 2 were used, the average accuracy was 93.33%. However, when k values of 3 and higher were employed, the average accuracy increased to 100%. The findings of this study demonstrate the potential efficacy of the fuzzy logic algorithm and k-NN as a classification tool for identifying different varieties of mango commonly found in the Philippines. The models employed in this study were found to be highly appropriate for the rapid and efficient identification of images of mango plant leaves. The models presented exhibit imperfections and have potential for enhancement in order to yield more precise outcomes. The results indicate that the kNN classifier exhibited higher accuracy compared to the fuzzy logic algorithm.

The study conducted by the authors [17] involved the development of a neural network model known as a Multi-Layer Perception (MLP). This model was designed to accurately classify the variety of mango based on an image of its leaf. Specifically, the research focused on the four main mango varieties found in the Philippines, namely Carabao, Pico, Pahutan, and Kachamita. A total of nine color features, nine textural features, and seven Hu moments morphological features were extracted from each leaf image sample utilizing various image processing techniques, including the automatic threshold method of segmentation, median filter, dilation, and erosion. The Multi-Layer Perception (MLP) consists of an input layer with 25 neurons, a hidden layer with 50 neurons, and an output layer with 4 neurons. The performance of the recognizer was evaluated using a dataset consisting of 40 leaf images. This dataset included 10 samples for each variety, with some of the samples being used during the training phase and others not. The test achieved a 96% accuracy rate.

In the aforementioned study [25], a convolution-free transformer model was developed and trained exclusively on the Imagenet dataset, resulting in a model that exhibits competitive performance. The participants were instructed on the operation of a singular computing device within a time frame of fewer than 72 hours. The reference vision transformer, which consists of 86 million parameters, demonstrates a top-1 accuracy of 83.1% when evaluated on ImageNet using a single-crop

approach, without the utilization of any external data. Furthermore, a teacher-student strategy specifically tailored to transformers was introduced by them. The effectiveness of this approach is contingent upon the utilization of a distillation token, which serves to facilitate the acquisition of knowledge by the student through focused engagement with the teacher.The token-based distillation method demonstrated a notable level of interest, particularly in cases where a convolutional neural network is employed as the instructor. As a consequence, they are able to present findings that are comparable to convolutional neural networks in terms of performance on the Imagenet dataset, achieving an accuracy of up to 85.2%. Moreover, their approach demonstrates promising results when applied to other tasks.
The utilization of GLCM and K-Nearest Neighbor (KNN) techniques was employed by the author of this study [26]. The Prototype method is employed in system development. The experiment involved conducting tests on a total of 60 mango leaves, which were divided into training data and test data in an 80:20 ratio. The accuracy of the results varied. The highest level of accuracy is achieved at K = 3 with a rate of 81%, utilizing a total of 6 features. Similarly, at K = 6, the accuracy drops slightly to 78% while employing 5 features. Lastly, at K = 7, the accuracy further decreases to 74% with the utilization of 4 features. The concept of authenticity refers to the quality or condition of being genuine, original, or true to its nature. In the context of state-of-the-art, it pertains to the most advanced or cutting-edge technology One notable distinction between the present study and prior research lies in the employed pre-processing technique, the selection of features, and the chosen classification methodology. In this approach, the image of the mango leaf is transformed into grayscale, followed by a subsequent feature extraction procedure. Subsequently, the outcomes of the feature extraction process will be subjected to classification through the utilization of the K-Nearest Neighbor technique. The system generates output that corresponds to the classification of mango leaves, including varieties such as Kweni, Lalijowo, and Madu.

Table 2.1: List of Literature Reviews

| Ref | Task | Classifier | Dataset | Accuracy |
|---|---|---|---|---|
| [1] | Automatic fruit and vegetable classification | N/A | Multi class fruit and vegetable image dataset | N/A |
| [2] | Shape based Features and Neural Network classifiers | neural network; computer vision | Mango leaf images dataset | 90% |
| [3] | Computer vision techniques in the agriculture and food industry | ANN | Grain, fruits, meat and fish dataset | |
| [4] | ImageNet Classification | CNN | 1.2 Million high resolution image | N/A |
| [5] | Mango Fruit Grading System | Computer vision | Image classification dataset | 100% |
| [6] | Large-Scale Image Recognition | CNN | Image classification dataset | N/A |
| [7] | Automatic classification of plants based | ANN | Different leaf image | 96% |
| [8] | Automatic classification of plants based on their leaves Mango Leaves by Using Artificial Intelligence | ANN | Image Capturing and Image Data Set | 96% to 98% |
| [9] | Deep Residual Learning for Image Recognition | CNN | image classification dataset | N/A |
| [10] | Plant Disease Detection | ViT | Defected plant from images of leaves | 98.61% |
| [11] | Mango Tree Varieties Based on Image Processing | N/A | Mango leaf image dataset | 78% |

| | | | | |
|---|---|---|---|---|
| [13] | Deep Learning With Depth Wise Separable Convolutions | CNN | ImageNet Dataset | N/A |
| [19] | Model Scaling for Convolutional Neural Networks | CNN | model scaling | 91.7% |
| [20] | Leaf Based Trees Identification | CNN | images of leaves | 99.40% |
| [21] | Mango Grading Using Image Processing | Computer vision, ML | Image classification dataset | 88.88% |
| [29] | Classifying Types of Mango Based on Leaf Images | CNN | Colored mango tree leaves dataset | N/A |
| [31] | Plant Disease Detection | CNN, Vit | Image classification dataset | 98.61% |
| [32] | Mangifera indica leaves using digital image analysis | CNN (R-CNN) | Correlation based Feature (CFS) Selection Data | 88.33% to 97% |
| [34] | Detecting Mango Diseases and Pesticide Suggestions | DenseNet169 | Defected mango leaves images | 97.81% |
| [35] | Breast Cancer Histopathological Images Classification | ViT-DeiT | histopathology image dataset | 98.17% |

Table 2.2: Comparison Table

| Study | Model | Accuracy |
|---|---|---|
| [2] | Neural network; computer vision | 90% |
| [5] | Computer vision | 100% |
| [7] | ANN | 96% |
| [10] | Vit | 98.61% |
| [19] | CNN | 91.7% |
| [21] | Computer vision, ML | 88.88% |
| Our paper | Vit | 97% |
| Our paper | VGG16 | 98.64% |
| Our paper | Inception V3 | 87.19% |
| Our paper | Resnet 50 | 97.9% |
| Our paper | Xception | 98.89% |
| Our paper | EffecientNetB3 | 94.42% |
| Our paper | MobileNetV2 | 98.10% |

# Chapter 3

# Datasets and Experimental Setup

## 3.1 Dataset in metric learning

We assessed our methodology using data we had collected ourselves. The dataset includes almost 14,000 images of 26 unique varieties of healthy mango leaves which is the foundation of our research.

### 3.1.1 Model training

When training a feature extraction network, it is usual practice to select certain categories of images from the dataset, divide these photos into validation sets, and then utilize the other images as training sets.

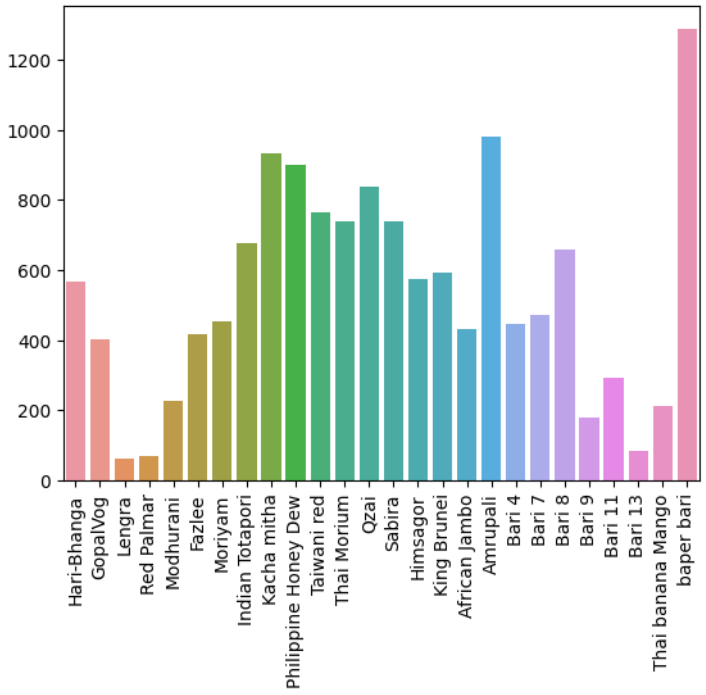



Figure 3.1: A chart of the number of datasets

## 3.2 Description of data

The process of implementing the deep learning and Vision Transformer (ViT) model began with a comprehensive and careful gathering of raw data from various locations in the lush landscapes of Bangladesh.
For the purposes of our research experiment, we collected Mango leaves from various varieties, specifically: African jumbo, Amrapali, Baaper Bari, Bari 4, Bari 7, Bari 8, Bari 9, Bari 11, Bari 13, Fazlee, Gopalvog, Harivanga, Himsagar, King Brunei, Lengra, Madhurani, Moriyam, Qzai, Red Palmer, Thai Banana Mango, Indian totapuri, Kacha Mitha, Philippine Honey Dew, Thai Moriyam, Sabira, and Tawani Red. The photographs in our possession were obtained from several sources, including SOAS Agro in Feni, Bangladesh Agricultural Research Institute in Rangpur, Brac Nursery in Demra, as well as several unidentified gardens in Rajshahi, Munshiganj, Tangail and Manikganj. The photographs were obtained using several high-resolution cell phone cameras and were positioned on a pristine white surface. The dataset that was gathered consisted of a minimum of 300 to 700 photos for each type of mango. The primary objective of this data collection expedition centered around the famed mango trees located in the Rajshahi Division, which are widely recognized for their remarkable flavor and captivating qualities. There are a total of 14,019 raw leaves. The depicted images exhibited a moderate distance and were characterized by a lack of sharpness. We tried to maintain a constant height of one and a half feet above the leaves, ensuring standardization of the entire image acquisition procedure. The photographs were taken during a period of clear weather conditions during midday, specifically between the hours of 11:00 a.m. and 3:00 p.m. Additionally, a few of the shots were captured under the illumination of linear light bulbs.

Table 3.1: Table of Data Collection

| | |
|---|---|
| Subject | Deep Learning |
| Specific subject area | Mango leaf variety classification |
| Type of data | Digital images of average size 128 × 128 pixels having RGB color in JPG format. |
| How the data were acquired | We examine a total of **26** distinct varieties of mango leaves that have a significant impact on mango trees. In order to gather data from several mango gardens throughout different regions of the country, four mango gardens in Bangladesh were chosen based on their dimensions and the diversity of tree species present. The leaf photographs were captured several days prior to the commencement of the summer season in 2023. The variety of the trees was repeatedly validated by agricultural experts.<br>Following the collection of the leaves, individual photographs of each leaf were obtained using a cell phone camera on a white background. Approximately 14,000 pictures were captured in all. |
| Description of data collection | Throughout the data collection process, utmost care and consideration were exercised to ensure the preservation of the mango trees and their blossoms. Therefore, the ideal time for leaf collecting was meticulously determined to be during the initial weeks of February and March. These months were deemed optimal to avoid any detrimental effects on the trees or their precious mango blooms.<br>Moreover, raw leaves were carefully processed through a complete cleaning procedure to remove any impurities, such as dust particles before being added to the dataset. The aim of this filtering process was to make the resulting deep-learning model more accurate and precise. With the raw leaves now properly prepared, a series of detailed photographs were captured, following a thorough drying process. These photographs were an essential foundation for the subsequent stages of the research, aiding in the training and optimization of the sophisticated deep learning and VIT model. |
| Data source location | The following mango orchards of Bangladesh are used for data collection:<br>1.SOAS Agro **(Feni)** 2.Bangladesh Agricultural Research Institute **(Rangpur)** 3.Brac Nursery**(Demra)** 4.Renowned mango gardens of **Rajshahi, Chapai Nawabganj** and **Munshiganj**. |

Table 3.2: Data Set Table

| **S/L** | **Classification Name** | **Number of Image** | **S/L** | **Classification Name** | **Number of Image** |
|---|---|---|---|---|---|
| 01 | Harivanga | 566 | 14 | Sabira | 738 |
| 02 | Gopal Vag | 402 | 15 | Himsagar | 575 |
| 03 | Lengra | 61 | 16 | King Brunia | 591 |
| 04 | Red palmar | 69 | 17 | African Jambo | 433 |
| 05 | Madhurani | 225 | 18 | Amrapali | 981 |
| 06 | Fazlee | 415 | 19 | Bari 4 | 447 |
| 07 | Moriyum | 455 | 20 | Bari 7 | 470 |
| 08 | Indian Totapori | 675 | 21 | Bari 8 | 659 |
| 09 | Kachamitha | 931 | 22 | Bari 9 | 178 |
| 10 | Philippines Honey Deu | 899 | 23 | Bari 11 | 293 |
| 11 | Taiwani Red | 764 | 24 | Bari 13 | 85 |
| 12 | Thai Moriyom | 738 | 25 | Thai Banana Mango | 212 |
| 13 | Qzai | 839 | 26 | Baper Bari | 1288 |

The above 3.2 table highlights our dataset which comprises nearly 14,000 high-resolution images of **26 unique** mango leaf varieties, meticulously collected from diverse locations in Bangladesh, serving as the **foundation for our research.**

# Chapter 4

# Proposed Methodology

The proposed methodology is initiated by preparing the entire raw dataset, followed by a thorough cleaning process to eliminate any images that exhibit blurriness. Following a systematic data processing approach, an embedding layer is constructed to facilitate the application of deep learning models, such as convolutional neural network (CNN) models. The process of picture acquisition involves the capture of leaf images through the utilization of various cell phone cameras. Picture pre-processing is a crucial initial stage in the classification of mango leaves. The completion of this step is crucial for the preparation of data for subsequent processing by Convolutional Neural Network (CNN) models. The preparation pipeline encompasses a series of meticulously selected methods aimed at enhancing features, expanding the dataset, and standardizing the images. The segmentation phase was utilized to isolate the leaf region and eliminate any extraneous surfaces. The subsequent re-processing stage included enhancing the acquired image, particularly focusing on rectifying any damaged portions. During the process of feature extraction, we obtained the leaf attributes of several categories in order to conduct texture analysis. The feature optimization process resulted in the identification of the most pertinent attributes for texture analysis while eliminating any extraneous features. As a result, we produced a dataset consisting of optimized features. The classification step involved the deployment of the LMT and KNN algorithms to discern between different varieties of leaves. The overall success of the model training process is influenced by the collective implementation of these phases.

**Components of detection**
1. VGG-16 Architecture 2. ResNet-50 Architecture 3. Inception V3 Architecture 4. EfficientNetB3 Architecture 5. MobileNetV2 Architecture 6. Xception Architecture 7. Vision Transformer Architecture

## 4.1 Data Pre-processing

Data pre-processing is a vital step when working with image data in Deep learning. Considering our dataset is completely self-collected, the orientation was variable since the photos were taken at various sizes with a white background. Before supplying data to deep learning models, we prepared it so that it could be read by machines. We worked on image resolutions, Augmentation Techniques like rotation and flipping, Gaussian Noise, Horizontal and Vertical Shearing, Gaussian Blur,

Color Jittering, Grayscale Conversion, Grid Distortion, Optical Distortion, Affine Transformation, and random brightness, and Rationale for Techniques like Enhancement Techniques (CLAHE), Augmentation Noise and Blur Techniques in order to implement CNN algorithms.

**Image Preprocessing**

In the pursuit of implementing powerful deep learning models, a crucial step involved mandatory image pre-processing before delving into the model implementation. The focus of this pre-processing endeavor was primarily directed toward Convolutional Neural Network (CNN) models, which are often utilized for transfer learning, as well as the innovative Vision Transformer (ViT) model. To cater to the specific requirements of these models, two distinct color scales, RGB and Gray, were considered.

For CNN models, such as VGG16, ResNet50, and Xception, it was observed that processing images in the RGB color scale yielded more accurate results. To prepare the data for the CNN models, an extensive range of pre-processing techniques were applied. These techniques encompassed a multi-step approach, including rotation, flipping, image enhancement, random brightness, and contrast adjustments, as well as the addition of random Gaussian noise. Furthermore, horizontal and vertical shearing, random Gaussian blurring, and random color jitter were also employed to augment the dataset's diversity.

To expand the dataset size, various transformation techniques such as random grayscale conversion, random grid shuffling, optical distortion, and random affine transformations were introduced. Finally, to standardize and facilitate model training, normalization was applied to the pre-processed images.

To expand the dataset size, various transformation techniques such as random grayscale conversion, random grid shuffling, optical distortion, and random affine transformations were introduced. Finally, to standardize and facilitate model training, normalization was applied to the pre-processed images.

The comprehensive data augmentation process proved to be highly effective, resulting in a significant increase in the dataset size from its original count of 13,898 to a more extensive collection after pre-processing the data containing 85,156 images. This augmented dataset played a pivotal role in enhancing the CNN models' ability to generalize and recognize patterns.

### 4.1.1 Image Resolution

#### Resolution 64

The following factors led to the resolution of 64x64 pixels being chosen for ResNet50, VGG16, MobileNetV2, and EfficientNetB3, ViT:
• Computing Efficiency: When compared to other models, these tend to have a higher number of parameters and computing needs. It strikes a compromise between maintaining important features and minimizing computational resources by downsampling the photos to 64x64 pixels. This resolution enables efficient training with minimal computing burden.
• Extraction of Features at Multiple Degrees of Abstraction: These models' architecture is built to extract features at various degrees of abstraction. While requiring less processing resources, lower-resolution photos can frequently capture important details. This is particularly crucial for models like the computationally efficient EfficientNetB3.

#### Resolution 128

The increased resolution of 128x128 pixels was used for InceptionV3 and Xception for the following reasons:
• Complicated Architectural Design: Inception V3 and Xception are constructed with complicated structures that can capture minute details and elaborate patterns. These models can better utilize their specialized structures with a higher resolution input, ensuring that no crucial information is lost during the initial processing steps. Inception V3 demands a minimum image resolution of 71x71, while Xception demands a resolution of 75x75.
• Handling Detailed Features: These models are especially good at dealing with intricate structures and features in photos. We guarantee that they have access to more thorough information by giving a higher-resolution input, which is necessary for precise leaf classification.
• Resistance to Variations: Leaves can have complex structures, patterns, and textures. The models are more resistant to changes in leaf shape and texture because they can more successfully capture these aspects with a higher-resolution input.
• Training Stability: Because they offer a more robust supply of data for the model to learn from, higher-quality photos frequently result in more steady training. As a result, training convergence may happen more quickly.

We attempt to optimize the performance of each model for the job of mango leaf classification by tuning the resolution to the particular characteristics and architectural details of each model.

### 4.1.2 Augmentation Techniques

A number of augmentation approaches are used to produce a diversified and reliable dataset. The risk of overfitting is decreased, and the model's capacity to generalize to new, untested data is improved by these controlled introductions of variables into the dataset. Each method has a distinct function:
•Rotation: By introducing variation in leaf orientations, this strategy makes the

model resilient to various angle viewpoints.
• Flip: By simulating mirror images, horizontal flipping broadens the dataset's diversity.
• Improvement: To improve contrast, Contrast Limited Adaptive Histogram Equalization (CLAHE) is used. This is very helpful for photos with different lighting.
• Random Brightness and Contrast: By exposing the model to various lighting environments, these tweaks help it become more adaptable to real-world circumstances.
• Gaussian Noise: Adding random noise makes the model more forgiving of probable noise in real-world photos.
• Vertical and Horizontal Shearing: These geometrical changes introduce deformations that mimic variations in leaf forms that occur naturally.
• Gaussian Blur: This method makes it simpler for the model to concentrate on important features by reducing high-frequency noise.
• Random color changes generate variances known as "color jittering," which can improve the model's ability to generalize to various color schemes.
• Grayscale Conversion: Grayscale conversion gives the model a more straightforward representation, potentially lowering computational requirements.
• Grid Distortion: By applying distortion to a grid, local deformations are introduced, further enhancing the dataset's diversity.
• Optical Distortion: By simulating optical distortions, the model is prepared for conceivable errors in picture acquisition.
• Affine Transformation: This combines shearing, scaling, rotation, and translation, and offers flexibility for dealing with leaves that are oriented differently.
• Justification for Techniques: Each pre-processing method is selected with a particular objective in mind.
• Light Enhancement Techniques (CLAHE): These techniques are used to deal with problems caused by erratic lighting conditions. The model can distinguish features better by increasing contrast.
• Augmentation Strategies To make sure the model can recognize leaves from multiple angles and orientations, rotation, flipping, and shearing imitate numerous real-world conditions. These methods also increase the dataset, lowering the chance of overfitting.
• Noise and Blur Techniques: The model can be trained to recognize characteristics even in noisy or blurry photos by exposing it to images with extra noise or blur.

### 4.1.3 Applying Augmentation

Each image in the collection is enhanced using various methods. This produces a varied collection of photos that provide the model marginally various viewpoints of the leaves.

**Normalization and Conversion**

The photos are normalized after augmentation. To provide numerical stability during model training, this entails scaling pixel values to the [0, 1] range. For learning to be consistent and efficient, this phase is essential.

### 4.1.4 Manually Removing the data

It was not entirely possible to collect all the data in the same lighting and environment which led to some of the images being corrupt. We manually removed those data for the CNN models to get a better output. In ViT, we used the "failed.map" feature to remove all the corrupt image data.

### 4.1.5 Pre-processed data saving

The pre-processed and enhanced images are stored as numerical arrays so that the CNN models can be trained right away.

### 4.1.6 Handling imbalance in dataset

As the dataset is collected all on our own, there are some imbalances in our data. Not all the 26 varieties contained the same amount of data which is why the augmentation process helped us to balance the data. Also, a leverage of using a pre-trained model is they are less sensitive to minority classes

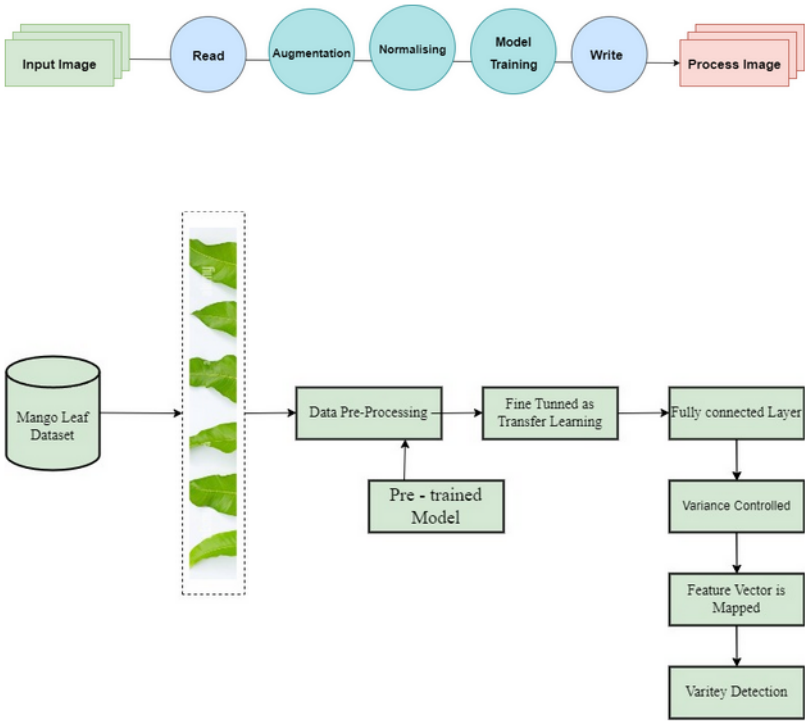



Figure 4.1: Workflow Diagram

## 4.2 Model Training

Images that have been pre-processed and optimized by resizing and normalization are loaded. For simplified training, label encoding assigns distinctive values. For generalization assessment, data is divided into training, validation, and test sets. It is essential to check that the data dimensions comply with the model's requirements. For multi-class classification, labels are prepared using one-hot encoding.
The categorization of mango leaves is complex, which is in line with the neural network design. It requires defining an optimizer, a loss function, and metrics. To reduce errors, training improves weights. On the validation set, generalized is evaluated.
Based on preliminary findings, hyperparameter fine-tuning may be taken into consideration. For a thorough performance assessment, the model is put to the test on fresh data. Effective mango leaf classification using deep learning depends on a thorough methodology that includes data preparation and review.

## 4.3 Model Description

Deep learning models have made substantial progress in fields including gaming, language interpretation, and picture analysis. The advantage of using these models is that they can acquire hierarchical data representations, gradually extracting more complicated attributes at each level. They have applications in a variety of fields, including banking, healthcare, and other areas. In order to increase overall precision, approaches for fine-tuning and improving are used with 6 CNN models: VGG16, ResNet50, Efficient Net B3, Inception V3, MobileNetV2, and Xception

### 4.3.1 Convolutional Neural Network Models

Mango leaf classification is a crucial task that makes use of cutting-edge deep learning architectures to correctly classify mango leaves. Six well-known models— VGG16, EfficientNetB3, MobileNetV2, Xception, InceptionV3, and ResNet-50—take front stage in this endeavor. Each model emphasizes its own advantages and architectural breakthroughs, ensuring a thorough examination of capabilities. Pre-trained models are expertly refined to the subtleties of mango leaf classification through painstaking modification.

#### CNN Architecture

This application is a perfect fit for the CNN model, a deep-learning technique that has shown exceptional success in picture identification tasks. The goal of this project is to create an effective and precise system for identifying mango varieties by automating the procedure with CNN-based ML algorithms. The process includes several stages, beginning with a thorough dataset collection made up of high-resolution pictures of numerous species of mango leaves. To improve the quality and standardize the images to reduce variability, pre-processing procedures are used. To

construct and test the model, the dataset is then split into training, validation, and testing sets and evaluate the CNN model effectively [10].

Multiple convolutional layers in the CNN architecture are created to effectively extract hierarchical information from the images of mango leaves. To reduce the loss function and increase classification accuracy, the model is trained using back propagation and optimization techniques. A thorough series of experiments is carried out utilizing various configurations of the model to assess the efficacy of the CNN-based mango variety detection system. To evaluate the efficacy and superiority of the suggested approach, the findings are contrasted with conventional manual identification techniques.

The model is also evaluated for generalized using previously unreleased data. The experimental findings show that the CNN-based mango variety recognition system outperforms manual identification techniques in terms of accuracy and efficiency. Even in difficult situations when there are minute leaf changes between closely related mango varieties, the deep learning approach greatly decreases the time and effort needed for classification while attaining a high degree of accuracy.

The suggested system not only helps farmers and academics quickly identify different mango types, but it also helps increase mango cultivation's overall productivity and sustainability. To further increase the system's robustness and adaptability in real-world circumstances, future research may concentrate on growing the dataset, improving the CNN design, and investigating the integration of other data sources. [28]

**Nonlinear activation function**

The following are the most typical nonlinear activation functions used in neural networks:

ReLU (Rectified Linear Unit): This function converts all negative values to zeros, adding non-linearity and enhancing the network's ability to learn intricate patterns. This method is advantageous for binary classification tasks because it converts the input to a value between 0 and 1. This function maps the input to a value between -1 and 1, which is beneficial for regression tasks.
Leaky ReLU: This function is similar to ReLU, but it permits a minor gradient when the unit is inactive, thereby preventing ReLU issues.
ELU (Exponential Linear Unit): This function is comparable to ReLU, but it produces more negative outputs, enabling the network to learn more complex features. In this variant of the rectified linear unit (ReLU) activation function, all negative values are converted to zero. Instead of returning zero for negative values, the Maxout activation function chooses the collection's highest positive value.

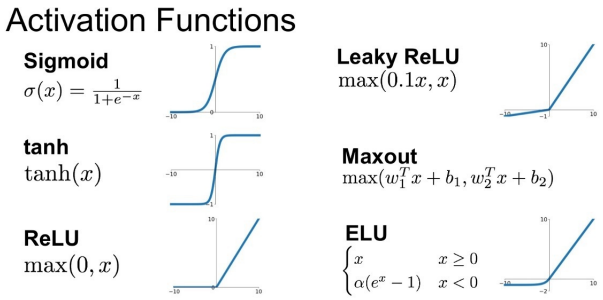



Figure 4.2: Different non-linear activation function.

### Pooling Layers

The feature maps produced by the previous convolutional layers are used by pooling layers to operate. The various facets or patterns in the input data are represented by these feature maps. A tiny window (commonly 2x2 or 3x3) is typically slid across the feature map during the pooling operation, with each point collecting the maximum or average value within the window. Max Pooling and Average Pooling are two popular varieties of pooling processes.
Max Pooling chooses the highest value from the window, highlighting the area's most salient feature. Through the reduction of spatial dimensions, this procedure aids in the preservation of significant local features. For instance, max pooling may be used to determine the most noticeable edge or texture feature within a region in an image identification task.
In contrast, Average Pooling determines the average value for the window. When performing tasks like semantic segmentation, it is typically used when a more comprehensive understanding of the area is sought.
In order for the network to be less sensitive to minute changes in the position of features inside an image, pooling layers are essential for establishing translation invariant. This trait is especially helpful in computer vision applications where it's crucial to recognize an object's presence regardless of its precise location.
The pooling layer assists in lowering the spatial dimensions of feature maps following convolutional processes, which is relevant to our thesis on the identification of the mango leaf variety. In addition to improving computing performance, this dimensional reduction also abstracts the most important aspects, making it simpler for later layers to acquire higher-level representations for precise categorization.
Pooling layer factors like window size and stride can have a big impact on how well the network performs. Therefore, it is crucial to carefully experiment with and analyze various pooling algorithms in order to establish the best configuration for our particular mango leaf variety identification task.
As our research develops, we will investigate how convolutional and pooling layers work together in CNNs, hoping to take advantage of their combined strength for reliable and accurate picture classification.

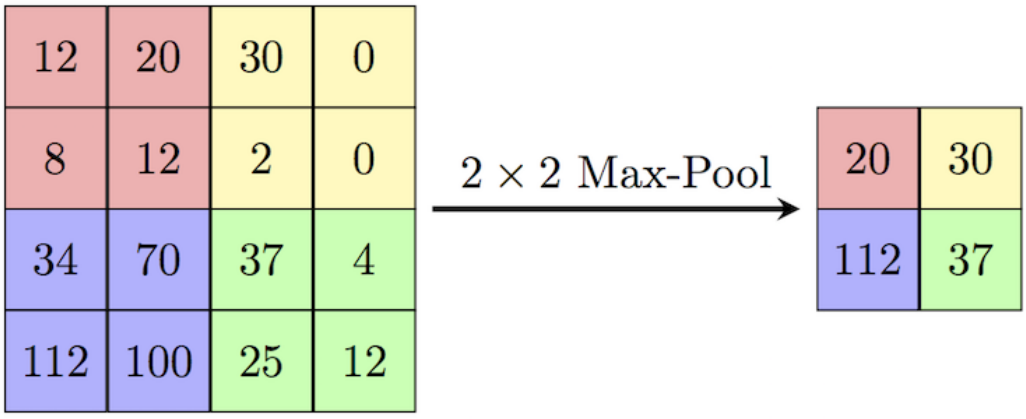



Figure 4.3: Sample diagram of pooling layer

**Flatten Layer**

In a neural network architecture, the Flatten layer acts as a link between the feature extraction layers, such as the convolutional and pooling layers, and the fully connected layers. Its main job is to turn the two-dimensional feature maps or arrays that the previous layers produced into a one-dimensional vector. In order to create a linear series, the elements in the arrays must be unstacked and rearranged.
The Flatten layer is crucial because it may transform the spatial data from feature maps into a form that can be handled by more established neural network layers, including fully connected layers. For classification or regression tasks, it essentially "flattens" the hierarchical representation of features learned by the earlier layers. The Flatten layer is essential in the context of image recognition, where CNNs are frequently utilized, as it transforms the spatial patterns and information retrieved by convolutional layers into a format suited for making predictions.

**Fully Connected Layer**

Every neuron, also known as a node, creates connections with every other neuron in the layer above and below it in the Fully Connected layer of a neural network. Each neuron in a completely connected layer receives input from every neuron in the layer below and transmits its output to every neuron in the layer above. This layer is also referred to as dense or feed-forward.
Fully linked layers' main function is to discover intricate, nonlinear correlations in the data. These layers provide for high-level feature representation and abstraction by allowing the network to integrate and weigh the features discovered in the preceding layers. Fully linked layers are particularly good at identifying broad patterns and formulating conclusions or predictions based on the data-extracted information.
In conclusion, the Fully Connected layer in a neural network architecture is in charge of learning complex relationships between these features and making final predictions or decisions, while the Flatten layer acts as a data transformation step by reducing multi-dimensional feature maps into a one-dimensional vector. Together, these layers are crucial elements for a variety of deep learning and machine learning tasks, such as object detection, natural language processing, and image categorization.

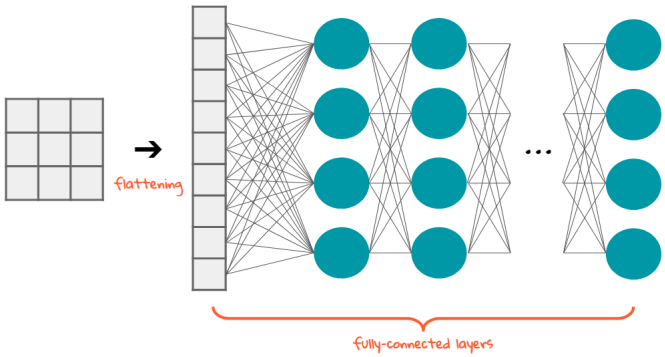



Figure 4.4: Diagram of flatten and fully connected layer

### VGG16

VGG16 is a 16-layer deep CNN model with various convolutional and fully linked layers. To utilize feature extraction capabilities from generic visual patterns, the model's weights are initialized with pre-trained weights on a large-scale dataset[23]. The VGG16 architecture is a well-known option for accurate mango leaf categorization. VGG16 is a well-known deep convolutional neural network (CNN) that has excelled at a number of image categorization tasks. We enhance a pre-trained VGG16 model to be excellent at classifying mango leaves. By removing its top layers, the model is set up to work only as a feature extractor. This VGG16 feature extractor is layered with a custom classifier that consists of Flatten, Dense, Dropout, and Output layers. The model can capture complex relationships and patterns in the data because of its structure. Categorical cross-entropy is used as the loss function and stochastic gradient descent (SGD) is used as the optimizer in the model. To dynamically change the learning rate during training, a learning rate reduction approach is implemented. With these elements in place, the model is iteratively trained on the training dataset to improve classification accuracy and reduce the categorical cross-entropy loss function. The VGG16 architecture is efficiently used in this method to classify mango leaves. Feeding the preprocessed mango leaf pictures into the network and iteratively tweaking the model's parameters to minimize the classification loss function is how the VGG16 model is trained. The optimization is carried out with the use of an efficient algorithm, such as stochastic gradient descent or Adam. The model learns to distinguish distinct features and patterns related to different mango types during the training process. The trained model's performance is then evaluated using metrics such as accuracy, precision, recall, and F1-score.VGG16's deep design enables it to learn intricate patterns and fine-grained traits necessary for differentiating closely related mango types. The technology detects mango varieties with excellent accuracy, beating existing manual methods significantly [6].

Here's an overview of the working process of VGG16:

**Layer Stacking:** VGG16 has 16 weight layers, including 3 fully linked layers and 13 convolutional layers. A deep network is created by stacking the convolutional layers on top of one another. A Rectified Linear Unit (ReLU) activation function, which introduces non-linearity, comes after each convolutional layer.
**Convolutional Layers:**Small 3x3 convolutional filters with a stride of 1 and'same' padding are used in the convolutional layers. These layers are in charge of spotting different edges, textures, and shape patterns in the supplied image. As we go further into the network, each convolutional layer has more filters, enabling the model to capture more complicated characteristics.
**Pooling Layers:** Max-pooling layers are included in VGG16 after a number of convolutional layers.The most crucial details are preserved while the feature maps' spatial dimensions are reduced thanks to max-pooling. The feature maps are typically downsampled using a 2x2 window and a stride of 2.
**Fully Connected Layers:** VGG16 has three fully connected layers after the convolutional and pooling layers. As a classifier, the fully connected layers combine the information discovered in earlier layers to generate predictions. The number of neurons in the last fully connected layer is often equal to the number of classes in the classification task (for example, 1,000 for ImageNet).
**Softmax Activation:** The softmax activation function is applied to the output of the last fully linked layer. Softmax provides the model's prediction for the input image by converting the raw scores into class probabilities.
**Training:** By changing its weights throughout training, the VGG16 learns to minimize a loss function using supervised learning. Cross-entropy loss is a typical loss function for classification applications. The model's weights are updated during training using backpropagation and optimization algorithms like stochastic gradient descent (SGD).
**Inference:** When performing inference, the VGG16 network performs forward propagation on an input image. The class with the highest probability is taken into consideration as the projected class for the input image after the final softmax layer generates class probabilities.
**Transfer Learning:** VGG16 is frequently used for transfer learning. For some jobs, pre-trained models that were developed from massive datasets like ImageNet can be fine-tuned using smaller datasets. The pre-trained model's weights are updated on the fresh dataset while the initial training's lessons are kept in mind while fine-tuning.

| Layer | | Feature Map | Size | Kernel Size | Stride | Activation |
|---|---|---|---|---|---|---|
| Input | Image | 1 | 224 x 224 x 3 | - | - | - |
| 1 | 2 X Convolution | 64 | 224 x 224 x 64 | 3x3 | 1 | relu |
| | Max Pooling | 64 | 112 x 112 x 64 | 3x3 | 2 | relu |
| 3 | 2 X Convolution | 128 | 112 x 112 x 128 | 3x3 | 1 | relu |
| | Max Pooling | 128 | 56 x 56 x 128 | 3x3 | 2 | relu |
| 5 | 2 X Convolution | 256 | 56 x 56 x 256 | 3x3 | 1 | relu |
| | Max Pooling | 256 | 28 x 28 x 256 | 3x3 | 2 | relu |
| 7 | 3 X Convolution | 512 | 28 x 28 x 512 | 3x3 | 1 | relu |
| | Max Pooling | 512 | 14 x 14 x 512 | 3x3 | 2 | relu |
| 10 | 3 X Convolution | 512 | 14 x 14 x 512 | 3x3 | 1 | relu |
| | Max Pooling | 512 | 7 x 7 x 512 | 3x3 | 2 | relu |
| 13 | FC | - | 25088 | - | - | relu |
| 14 | FC | - | 4096 | - | - | relu |
| 15 | FC | - | 4096 | - | - | relu |
| Output | FC | - | 1000 | - | - | Softmax |

Figure 4.5: Vgg16 Features

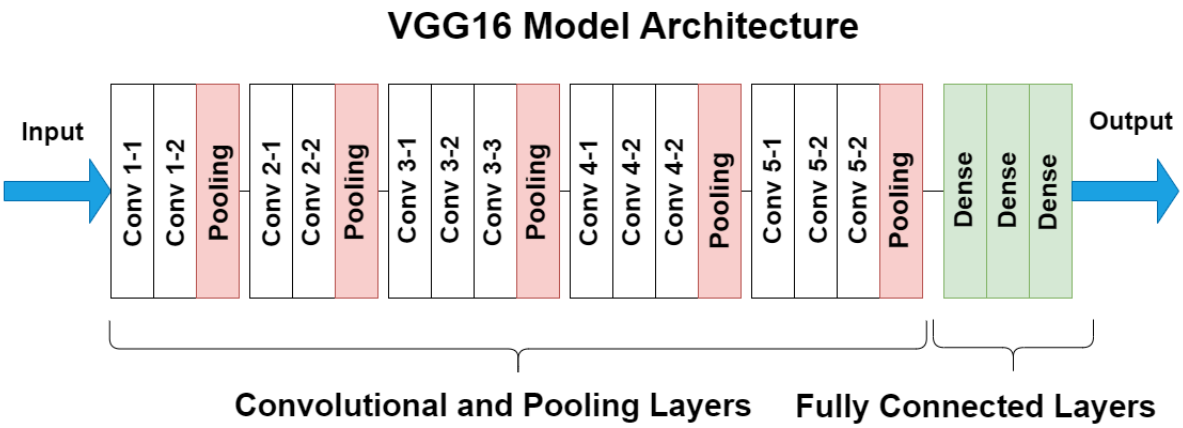



Figure 4.6: Vgg16 Model Architecture

### EfficientNetB3

EfficientNetB3 is a cutting-edge deep learning model architecture introduced in 2019 by Google Research's Mingxing Tan and Quoc V. Le.EfficientNetB3's primary innovation is its compound scaling strategy, which balances model depth, width, and resolution to outperform conventional models with similar parameter counts.
To classify mango leaves, we use the EfficientNetB3 architecture, which is renowned for striking a balance between model size and precision. The effective scaling strategy used by this model, which consistently modifies network dimensions for en-

hanced performance, makes it particularly effective at picture classification tasks. To specifically tailor the features obtained by EfficientNetB3 for this task, a bespoke classifier is painstakingly built. It has elements like Output layers, Dense, Dropout, and Batch Normalization. The Adamah optimizer is used to build the model, and categorical cross-entropy is used as the loss function. To dynamically change the learning rate during training, a learning rate reduction approach is implemented. The model is iteratively trained on the training dataset with these elements in place to improve classification accuracy. An effective base for precise mango leaf classification is created by combining the feature extraction capabilities of EfficientNetB3 with the custom classifier.

The design of EfficientNetB3 is based on the Convolutional Neural Network (CNN) framework, which has proven to be incredibly effective in computer vision tasks including object detection and image categorization. Conventional CNN designs, however, have a trade-off: either they are shallow, computationally affordable, but have a limited capacity for representation, or they are deep, powerful, but computationally intensive, making them challenging to deploy on devices with limited resources [15].

EfficientNetB3 creates a balance between efficiency and performance by scaling these three components together. On many test datasets, such as ImageNet, it regularly outperforms earlier CNN architectures while requiring fewer parameters and less processing. This makes EfficientNetB3 particularly appealing for implementation on low-resource devices such as smartphones and embedded systems.

$$depth : d = \alpha^{\phi} \tag{4.1}$$

$$width : w = \beta^{\phi} \tag{4.2}$$

$$resolution : r = \gamma^{\phi} \tag{4.3}$$

$$s.t.\alpha.\beta^2.\gamma^2 \approx 2 \tag{4.4}$$

$$\alpha \geq 1, \beta \geq 1, \gamma \geq 1 \tag{4.5}$$

The working process of EfficientNetB3 can be broken down into several key steps:

**Base Architecture Selection:** The EfficientNetB3 family consists of several models, such as EfficientNetB3-B0, EfficientNetB3-B1, EfficientNetB3-B3 and so on, with varying model sizes. The selection of the base architecture is the starting point. Smaller models are less resource-intensive but may have reduced capacity, while larger models can capture more complex features but require more computation.

**Scaling Depth, Width, and Resolution:** The central concept of EfficientNetB3s is to scale the network's depth (number of channels), width (number of channels), and image resolution in a manner that optimizes performance and efficiency. EfficientNetB3s use a method of compound scaling in which depth, width, and resolution are increased simultaneously while maintaining a constant ratio.

**Feature Extraction:** EfficientNetB3s employ convolutional layers to extract features. In order to detect low-level and high-level features, these layers apply a series of learned filters to the input image or feature maps.Utilized frequently are depth-separable convolutions, which are computationally efficient and reduce the number of parameters.

**Feature Aggregation:** Following the extraction of features, the network aggregates information across multiple dimensions and layers. This is typically done through skip connections or skip connections combined with global average pooling (GAP).
**Efficient Attention Mechanisms:** To improve feature representation, some variants of EfficientNetB3s employ efficient attention mechanisms such as SE (Squeeze-and-Excitation) or efficient self-attention. These mechanisms enable the network to prioritize significant characteristics while suppressing irrelevant ones.
**Strategy for Training:** Typically, EfficientNetB3s are trained using standard deep learning techniques, such as stochastic gradient descent (SGD) or Adam, with appropriate learning rate schedules. Additionally, data augmentation, regularization techniques, and label normalization can be used to enhance generalization.
**Fine-Tuning and Transfer Learning:** Transfer Learning and Fine-Tuning EfficientNetB3s can be fine-tuned for particular tasks using transfer learning. Models trained on massive datasets such as ImageNet can be modified to perform well on task-specific datasets.
**Inference and Deployment:**Once trained, EfficientNetB3s can be used for inference on new data and deployment. They are frequently deployed on hardware accelerators such as GPUs and TPUs for real-time prediction efficiency.
**Model Evaluation:** The efficacy of the trained EfficientNetB3 model is evaluated using appropriate evaluation metrics, such as precision, recall, F1-score, and mean average precision (mAP).
**Model Tuning (Optional):** Depending on the specific task and performance requirements, hyperparameter tuning and architectural modifications may be used to further optimize the model.

The accuracy rate formula for an EfficientNetB3 convolutional neural network (CNN) is typically calculated using the following formula:

$$AccuracyRate(\%) = (NumberofCorrectPredictions/TotalNumberofPredictions)100 \tag{4.6}$$

In this formula: Number of Correct Predictions refers to the count of correctly classified examples in your dataset. These are the predictions made by the model that match the ground truth labels.Total Number of Predictions is the total count of predictions made by the model on your dataset.

### ResNet-50

ResNet-50 is a deep CNN model with 50 layers that uses residual connections to successfully train very deep networks. These residual connections improve network information flow by minimizing the vanishing gradient problem and enabling better gradient propagation [9].
The ResNet-50 model is trained by feeding the preprocessed mango leaf images into the network. The model's parameters are iteratively altered during training using an efficient optimization method, such as stochastic gradient descent or Adam, to reduce the classification loss function.

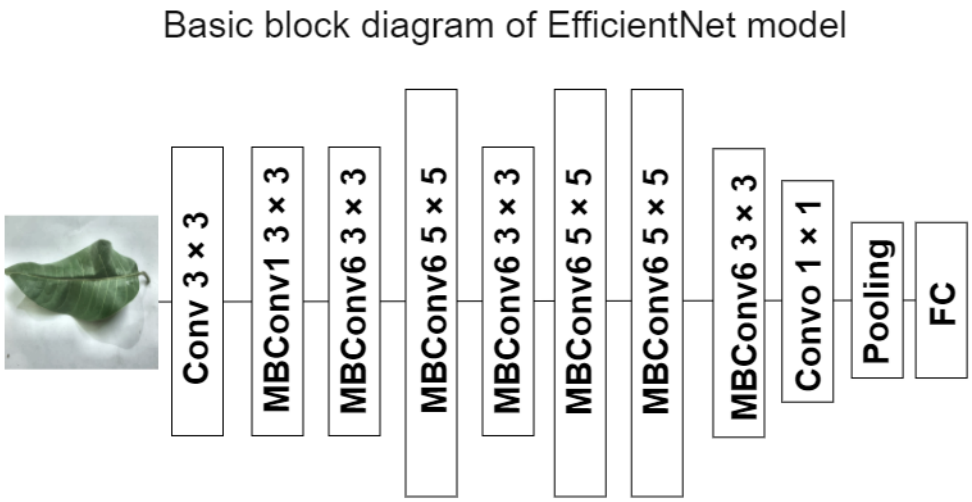



Figure 4.7: Basic Block Diagram of EfficientNetB3 Model

The ResNet-50 architecture proves to be a potent resource in our pursuit of precise mango leaf classification. ResNet-50 is renowned for its efficiency in deep learning applications, especially image identification, and it stands out for its capacity to build very deep neural networks. The vanishing gradient problem is mitigated by ResNet's inclusion of residual blocks, which enable the network to learn residual functions. Due to this innovative method, detailed characteristics can be captured successfully by layering on top of each other. An output layer for multi-class classification is built into a bespoke classifier, which also includes densely linked layers and a flatten layer. The categorical cross-entropy loss function, the accuracy evaluation metric, and the Adam optimizer are all used in the model's construction. The incorporation of a learning rate reduction technique enhances the model's convergence and overall performance. The model is iteratively trained on the training dataset using these components in an effort to decrease the categorical cross-entropy loss function and boost classification precision. This approach for categorizing mango leaves successfully takes advantage of the ResNet-50 architecture.

The ResNet-50 architecture based on leaf analysis for mango variety detection is an extremely promising method. The established approach provides a reliable and fast method of identifying mango varieties, benefiting the horticultural industry and agricultural research significantly. To improve the system's accuracy and resilience, further study may include investigating various deep learning architectures, fine-tuning hyperparameters, and incorporating more data sources, such as multispectral photos.

K. Simonyan and A. Zisserman and then Szeged and W. Li have mentioned in their paper about the importance of network and depth in how the result changes significantly. A graph for loss error count with respect to irritations of the layers can give us an idea of how the result changes just by stacking the new layers.

When we delve into the training of deeper neural networks, we encounter a challenge known as the 'degradation problem.' This issue becomes apparent when, as network depth increases, the accuracy initially improves but eventually plateaus and even starts to decline rapidly. Remarkably, these accuracy declines are not attributable to overfitting **tan2019EfficientNetB3**, and instead, the inclusion of additional layers in the model exacerbates the training error. Figure 1, as demonstrated in our

experiments, provides a clear illustration of this phenomenon.
To address this degradation problem, the deep residual learning framework was introduced as a solution, as elucidated in the paper authored by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun [14]. In this context, the author endeavors to depict H(x) as the desired mapping and employs the mapping equation $F(x) = H(x)x$ to construct nonlinear layers through mapping. Consequently, the transformation of $F(x) + x$ seeks to restore the original mapping, thus mitigating the effects of the degradation problem.

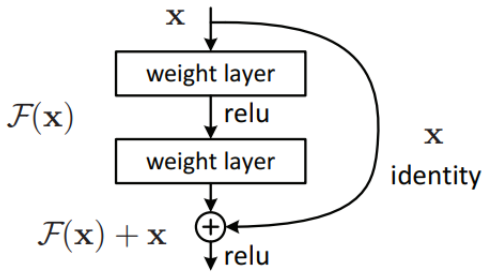



Figure 4.8: Residual learning: a building block

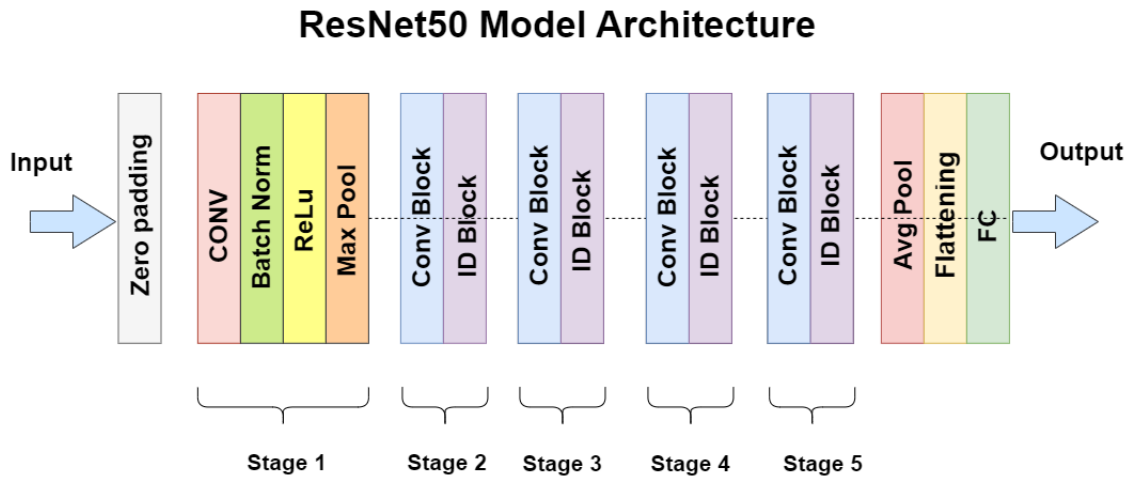



Figure 4.9: ResNet50 model architecture

### The InceptionV3

The InceptionV3 architecture is chosen for mango variety detection. Multi-scale filters (Inception modules) are a feature of Inception-v3 that allows the network to record features at various spatial resolutions within the same layer. These Inception modules make it possible to efficiently and thoroughly extract features, which strengthens the model's capacity for representation [12].
Additionally, during training, Inception-v3 adds supplemental classifiers to intermediate layers. These classifiers solve the issue of disappearing gradients and stabilize the learning process by supplying additional supervision signals and promoting gradient flow.
The pre-processed photos of mango leaves are fed into the network to train the Inception-v3 model. Using effective optimization algorithms like stochastic gradient descent or Adam, the model's parameters are iteratively changed to minimize the classification loss function.

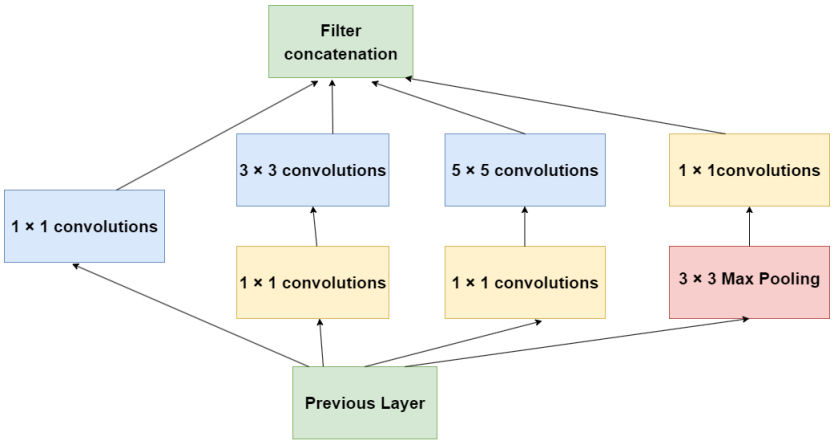



Figure 4.10: Inception module with dimension reductions

Factorized Convolutions: Inception V3 employs a technique known as 'factorized convolutions' to achieve a reduction in both parameter count and computational demands within its convolutional layers. Furthermore, it continually assesses the network's effectiveness.
The concept behind factorized convolutional involves breaking down a conventional convolution operation into two smaller convolution operations, each with a reduced number of filters. In the context of Inception V3, this method dissects a standard 3x3 convolution into two smaller components: a 1x3 and a 3x1 convolution. This strategic decomposition enables the network to utilize fewer filters, thereby diminishing the overall parameter count, all while capturing the same types of features as a traditional 3x3 convolution.
Additionally, factorized convolutions are extended to dissect a standard 5x5 convolution into two smaller counterparts: a 3x3 and either a 5x1 or a 1x5 convolution. This technique further streamlines the network by employing fewer filters and reducing the parameter count, while retaining the capability to detect the same feature characteristics as a standard 5x5 convolution.
Utilizing smaller convolutions is another strategy embraced by Inception V3. For instance, it incorporates 1x1 convolutions in some of its branches. These 1x1 convolutions are positioned to diminish the input's dimensionality before it undergoes

processing by larger convolutional filters. This dual benefit enables the network to focus on specific features, concurrently diminishing the number of parameters and lowering computational requirements.
In some branches of the Inception modules, a 1x1 convolution is strategically placed as a bottleneck layer before the 3x3 and 5x5 convolutional layers. The purpose of this strategic placement is to minimize the number of input channels and parameters before the data engages with the broader filters. As a result, this methodology assists in mitigating overfitting by reducing the number of parameters and enabling the neural network to focus on higher-level abstract characteristics.

In summary, the architecture of Inception V3 strategically incorporates a combination of smaller convolutional filters, factorized convolutions, and larger convolutional filters to extract salient features from input images. By employing convolutional filters with fewer dimensions, the network effectively reduces input dimensionality and keeps the parameter count in check. These measures collectively contribute to heightened computational efficiency and improved accuracy of the network.

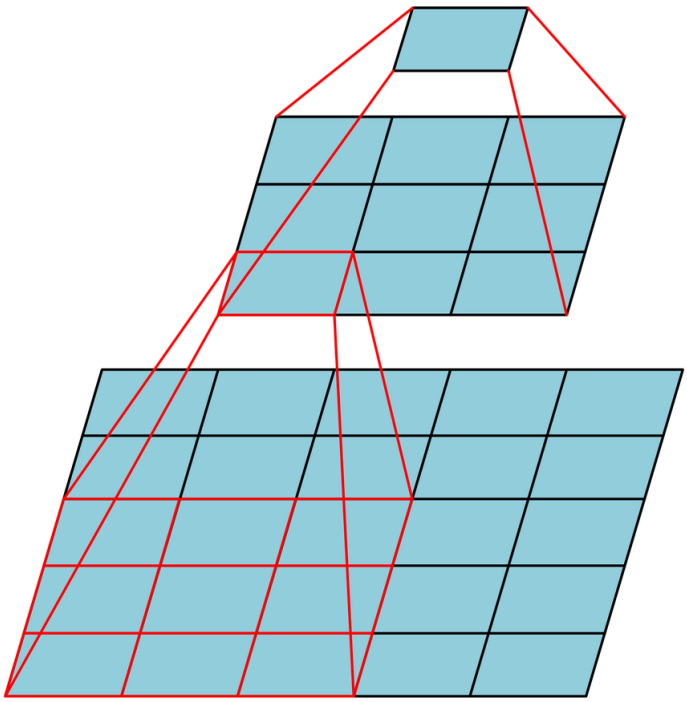


Figure 4.11: Mini-network replacing the $5 \times 5$ convolutions

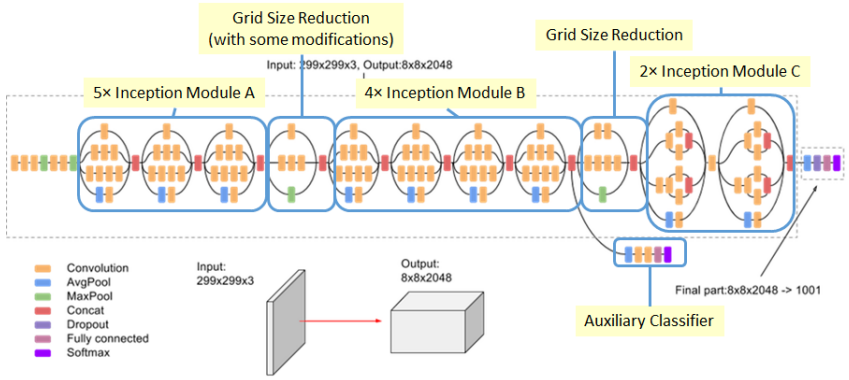



Figure 4.12: Inception V3 Architecture

### The MobileNetV2 Architecture

The MobileNetV2 architecture is selected for mango variety detection. The pre-processed pictures of mango leaves are fed into the MobileNetV2 model to train it. During the training process, the model learns to identify distinctive traits and patterns unique to various mango types. Using effective optimization algorithms like stochastic gradient descent or Adam, the MobileNetV2 model undergoes repeated modifications to reduce the classification loss function by adjusting its parameters **howard2017MobileNets**.

We choose the MobileNetV2 architecture in our effort to classify mango leaves because of its effectiveness and adaptability for tasks with limited resources. Given its popularity and minimal computational demands, MobileNetV2 is a great option for image categorization on devices with constrained resources. An excellent basis for learning features from images is provided by the initialization of the architecture with pre-trained weights from the ImageNet dataset. The pre-trained MobileNetV2 feature extractor is followed by a bespoke classifier that consists of a single dense layer and an activation feature for SoftMax. The model is constructed using stochastic gradient descent (SGD), and the loss function is categorical cross-entropy.

To optimize the model's convergence and performance, a learning rate reduction method is used. With these components in place, the model is iteratively trained on the training dataset, with weights being updated based on prediction errors to reduce the categorical cross-entropy loss function. This method makes use of MobileNetV2's effectiveness to deliver an efficient and resource-conserving solution for the categorization of mango leaves.

The experimental findings show how well the MobileNet architecture distinguishes between different mango kinds based on their leaves. MobileNet's lightweight and effective architecture makes it capable of processing images quickly, making it appropriate for real-time applications on devices with limited resources. Its efficiency and small size make it a realistic alternative for deployment in settings with constrained processing resources, despite the fact that it might have slightly poorer accuracy compared to more complicated models.

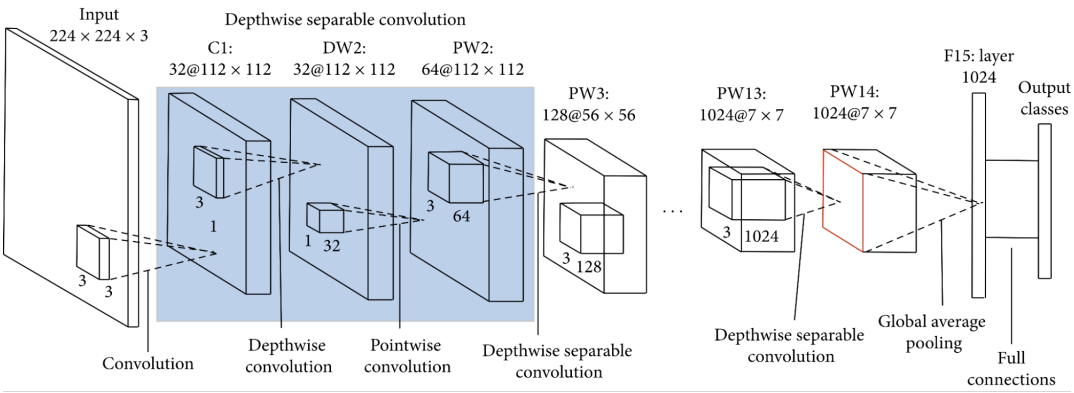



Figure 4.13: Dense-MobileNetV2 Model

These networks are known for their compact size and computational efficiency while maintaining good performance on various computer vision tasks. Here's an overview of the working process of MobileNetV2:
Depthwise Separable Convolution: The main innovation of MobileNetV2 is the use of depthwise separable convolution, which substitutes standard convolutions used in larger networks such as VGG or ResNet.Depthwise separable convolution is made up of two steps: depthwise convolution and pointwise convolution. Depthwise convolution applies a single filter to each input channel on their own followed by pointwise convolution, which uses 1x1 convolutions to merge the depthwise convolution outputs. This separation drastically reduces the number of parameters and computations, making MobileNet highly efficient.
Width Multiplier and Resolution Multiplier: MobileNet introduces two hyperparameters, the width multiplier $\alpha$ and the resolution multiplier $\rho$, which allow you to control the model's size and computational cost. The width multiplier scales the number of channels in each layer. A smaller width multiplier results in a narrower network with fewer parameters. The resolution multiplier scales down the input image resolution. A smaller resolution multiplier reduces input image dimensions and further reduces computational requirements.
MobileNet comes in various versions, such as MobileNetV1, MobileNetV2, and MobileNetV3, each with architectural improvements.MobileNetV1 introduced depthwise separable convolutions.MobileNetV2 added inverted residual blocks with linear bottlenecks and skip connections, enhancing performance.MobileNetV3 introduced features like a non-linear activation function called the Swish activation and network architecture search to optimize model design.
Training and Fine-Tuning: MobileNets are trained using standard deep learning techniques, including gradient descent and backpropagation, on large datasets such as ImageNet, a method for turning on SoftMax. The model is built with stochastic gradient descent (SGD), and category cross-entropy is used as the loss function.

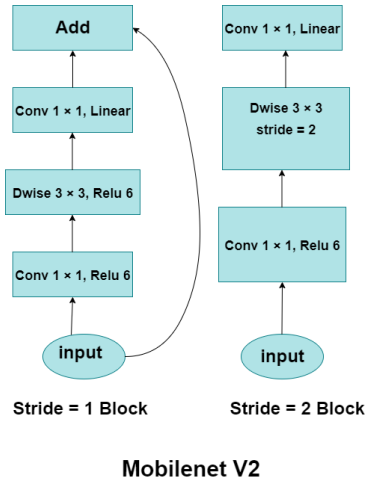



Figure 4.14: Workflow of MobileNet V2

Inference: During inference, MobileNet takes an input image, scales it according to the resolution multiplier (if used), and passes it through the network. The depthwise separable convolutions efficiently extract features, and the output is typically passed through a classifier (e.g., a fully connected layer with softmax) for classification tasks or a regression layer for object detection tasks.
Deployment on Mobile and Embedded Devices: MobileNet's compact size and computational efficiency make it ideal for deployment on mobile phones, IoT devices, and embedded systems. Various deep learning frameworks and libraries support MobileNet for deployment, such as TensorFlow Lite and PyTorch Mobile.

Any deep neural network model's inference time formula, including MobileNetV2, is often dependent on how many FLOPs (Floating-Point Operations) the model performs on input during its forward pass. The following formula can be used to estimate the inference time (in seconds):

$$InferenceTime(seconds) = NumberofFLOPs/FLOPspersecond \tag{4.7}$$

Number of FLOPs: This gauges how many floating-point operations are involved in processing one input through the model. In the case of MobileNetV2, it is determined by adding the FLOPs for all network levels. The input and output dimensions, filter size, and layer type—such as convolution, depthwise convolution, or completely connected—all affect each layer's FLOPs. FLOPs per second**: The processing speed of the hardware on which the model is executing is indicated by this number. The hardware's capacity is commonly expressed in giga-FLOPs per second (GFLOPs/s), which represents how many FLOPs it can execute in a single second.
A deep learning framework can be used like TensorFlow or PyTorch, which frequently offers built-in functions or utilities to profile the inference time for your model on a certain hardware setup, to measure the real inference time on a particular device.

$$\hat{x} = \frac{x - E[x]}{\sqrt{\text{Var}[x] + \varepsilon}} \tag{4.8}$$

$$y = \gamma \times \widehat{x} + \beta \tag{4.9}$$

$$y = \gamma \frac{x - E[x]}{\sqrt{\text{Var}[x] + \varepsilon}} + \beta \tag{4.10}$$

$$y = \frac{\gamma x}{\sqrt{\text{Var}[x] + \varepsilon}} + \left( \beta - \frac{\gamma E[x]}{\sqrt{\text{Var}[x] + \varepsilon}} \right) \tag{4.11}$$

### Xception

The deep convolutional neural network architecture known as Xception—short for "Extreme Inception"—was first presented by François Chollet in the 2017 publication titled "Xception: Deep Learning with Depth Wise Separable Convolutions" [13]. In order to reduce computational complexity and the number of parameters in the network while retaining high accuracy, the depth-wise separable convolutions concept, on which the Xception architecture is based, was developed. This is done by using depthwise separable convolutions in place of conventional convolutions. These convolutions consist of a depthwise convolution followed by a pointwise convolution. Image categorization is one computer vision problem where this design has been effective.

We use the XceptionNet architecture, well-known for its efficiency in image recognition applications and depth-wise separable convolutions, for mango leaf classification. This architecture makes use of depth-wise separable convolutions, which makes it possible to use computations and parameters more effectively. This makes it especially suitable for activities requiring less computational power. Pre-trained weights from the ImageNet dataset are used to provide a robust baseline for feature learning from pictures. A bespoke classifier made up of a single dense layer and a SoftMax activation function is added to the pre-trained XceptionNet feature extractor. The model is constructed using Stochastic Gradient Descent (SGD), with accuracy as the evaluation metric and categorical cross entropy as the loss function. In order to optimize the model's convergence and overall performance, a learning rate reduction method is used to dynamically alter the learning rate during training. On the training dataset, the model is iteratively trained with the intention of reducing the categorical cross-entropy loss function and increasing classification precision. This method combines a bespoke classifier with XceptionNet's effectiveness to deliver a reliable and resource-conserving solution for mango leaf classification.

The key to Xception's invention is its effective use of depth-wise separable convolutions, which enables it to effectively capture complicated information. When compared to conventional convolutional neural networks, Xception dramatically lessens the computational load by separating spatial and channel-wise convolutions. This architecture enables Xception to be more parameter-efficient while still achieving state-of-the-art performance on picture classification challenges. Since Xception can learn complex representations from incoming data thanks to depth-wise separable convolutions, the computer vision community has adopted it widely. This model is

a wise choice for a number of deep learning applications due to its adaptability and balance between accuracy and computational efficiency.

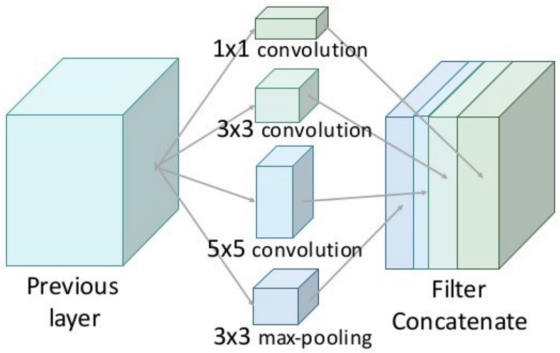



Figure 4.15: Architecture of Xception Model

Here's an overview of the working process of Xception:

**1. Depthwise Separable Convolution:** The usage of depthwise separable convolution, which is a more efficient variation of traditional convolutions, is a key advancement in Xception. It is divided into two steps which are depthwise convolution and pointwise convolution. Here, depthwise convolution applies a single filter to each input channel separately, lowering computational complexity when compared to ordinary convolutions that work on all channels simultaneously. In addition, to generate the final feature maps, pointwise convolution uses 1x1 convolutions to merge the results of depthwise convolution.
**2. Separable Convolutions:** Xception employs a series of these separable convolutions in its architecture, creating a deep and efficient network. By reducing the number of parameters and computation required in each layer, separable convolutions significantly contribute to Xception's efficiency.
**3. Skip Connections:** Xception incorporates skip connections, similar to those found in residual networks (ResNets). These skip connections help mitigate the vanishing gradient problem, allowing for more effortless training of very deep networks.
**4. Fully Convolutional Architecture:** Xception is designed to be fully convolutional, meaning it can process input images of varying sizes without the need for fully connected layers. This flexibility makes it well-suited for tasks like object detection and image segmentation, where input dimensions can vary.
**5. Training and Transfer Learning:** Xception is trained using standard deep learning techniques, such as stochastic gradient descent (SGD), backpropagation, and weight initialization. Pre-trained Xception models on large datasets like ImageNet can be fine-tuned for specific tasks using transfer learning, which helps boost performance on smaller, task-specific datasets.
**6. Inference:**During inference, Xception takes an input image and passes it through its deep network, applying the depthwise separable convolutions and skip connections to extract features. In image classification applications, the concluding layers of Xception architecture often include a global average pooling layer and a softmax classifier.

**7. Deployment and Frameworks:** Xception models can be deployed on various platforms using popular deep learning frameworks like TensorFlow and PyTorch. Deployment options include edge devices, cloud servers, and embedded systems.

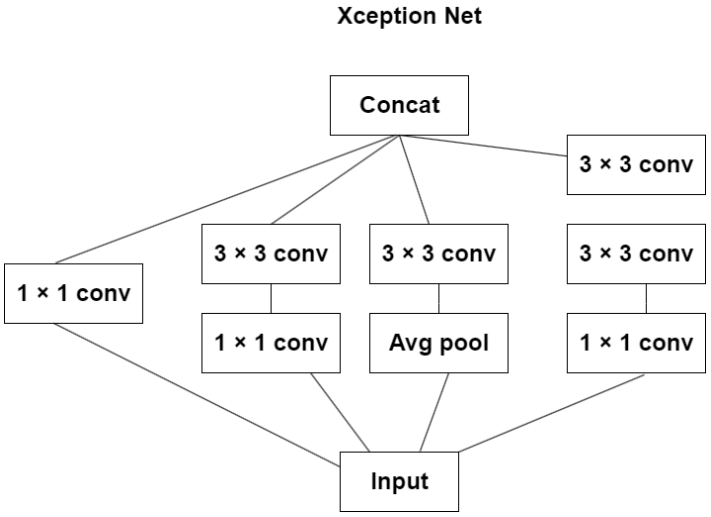



Figure 4.16: The filter bank outputs on the Xception modules have been increased

### 4.3.2 ViT

The Transformer architecture, which was initially created for natural language processing, is expanded into the field of computer vision by Vision Transformers (Dosovitskiy et al., 2021). The prevalent understanding that pixel-level convolutions are the most efficient method of processing images is called into question by this change. The way visual information is processed has fundamentally changed because of ViTs, which instead consider images as collections of discrete, non-overlapping patches.

The concept of self-attention, a process that enables the model to develop associations between various patches, is at the core of Vision Transformers (Dosovitskiy et al., 2021). This paradigm shift makes it possible for ViTs to effectively capture distant dependencies, contextual data, and complex patterns inside images. ViTs display impressive versatility, adaptability, and scalability across a wide spectrum of visual identification tasks by processing images as sequences of tokens.

The debut of Vision Transformers broadens the scope of computer vision study and application. In this thesis, we set out on an adventure to investigate and assess the viTs' capabilities, concentrating on their potential in the field of mango leaf variety identification. We seek to identify the advantages and disadvantages of ViTs and add to the expanding body of information about this game-changing technology by methodical experimentation, analysis, and comparison with conventional convolutional neural networks. We expect that as we delve further into this research, we will learn important lessons that not only enhance the science of computer vision but also present workable solutions to problems associated with image classification

in the real world.

The work process for Vision Transformers (ViT) in the context of our thesis involves several key stages:

**1. Model Selection:** Choosing the Vision Transformer architecture to be used (e.g., ViT-base, ViT-large) and deciding on specific hyperparameters such as learning rates, batch sizes, and training epochs.

**2. Built the model:** Divided the data randomly into 3 sections for training, testing, and validation. The ratio was 70% for trains, 10% for validation, and 20% for testing. We used two parameters called x-train and y-train. X-train defines the data whereas y-train defines the data label. Additionally, the transformer blocks generate a [batch-size, num-patches, projection-dim] tensor, which is then processed by a classifier head using softmax to provide the final class probabilities.

MLP Layer: The MLP layer is part of the feedforward neural network within the Transformer. It typically consists of two fully connected (dense) layers with an activation function applied in between. This MLP layer is applied independently to each patch's embedding.
Patch Encoder: A Vision Transformer's patch encoder divides raw input images into fixed-sized patches, flattens and embeds the patches into a lower-dimensional representation, and encodes the spatial relationships of the patches with position encodings. This initial stage of image processing converts the image to a format that is suitable for subsequent Transformer-based layers. The ViT model is capable of capturing both local as well as global features, thus making it suitable for a variety of computer vision applications. The first fully connected layer (often called the "hidden layer") increases the dimensionality of the patch embeddings, allowing for richer feature representations.
An activation function, such as the GELU (Gaussian Error Linear Unit) activation or ReLU (Rectified Linear Unit), is applied after the first layer. The second fully connected layer reduces the dimensionality back to the original embedding dimension.

**3. Training:** Initializing the ViT model with random weights and training the model on the training dataset using a suitable optimization algorithm. Then, monitor training progress by tracking loss and accuracy on the validation set.
ViT Classifier:
The Contextual Embeddings of the ViT model are processed and refined by the earlier layers of the model, the Transformer architecture incorporates key components such as the Self-Attention Mechanism and Feedforward Neural Networks.
These layers take in the contextual information of the input image and create a set of Contexted Embeddings (Ceilings) that represent both the local and global characteristics of the images. The Global Token, commonly referred to as the CLS Token, is pre-programmed into the sequence of Ceilings. The CLS Token provides a comprehensive overview of the image and provides useful information for prediction. The Classification Head is composed of one or multiple Fully Connected (Dense) Layer(s) that accept the CLS Token as input. Depending on the model

architecture, the Classification Head can be further activated with GELU or ReLU functions.
The ViT classifier is composed of one or more densely connected layers that accept the CLS token as an input. Depending on the architecture of the model, activation functions such as GELU or ReLU may be used. The output layer is typically composed of a number of units that correspond to the number of class classes in the task. For example, each unit in an image classification corresponds to a particular class label. This output layer is used to generate class probability scores, which indicate the likelihood that the input image belongs to each class. During inference, the highest probability score of the class is chosen as the predicted class, and during training the model is trained with labeled data using loss functions such as categorical cross-epoch. Ultimately, the model's weights, which include the classifiers as well, are adjusted via the process of backpropagation. This is done with the objective of minimizing the loss and improving the model's ability to accurately classify data.

**4. Model Run:** Firstly, We applied the Adam optimizer to adapt the learning rates. Then we compiled the model .

**5. Evaluation:** Assessing the trained ViT model's performance on the test dataset using evaluation metrics like accuracy, precision, recall, and F1-score and comparing the ViT model's performance with other models, such as CNNs, to evaluate its effectiveness for mango leaf variety identification.

**6.Visualization:** Visualize model predictions and any intermediate representations (e.g., attention maps) to gain insights into how the ViT processes mango leaf images.

**7.Hyperparameter Tuning:** Experimenting with different hyperparameters to optimize the ViT model's performance further.

**8.Documentation and Reporting:** Summarizing findings in your thesis, discussing the strengths and weaknesses of the ViT model for mango leaf variety identification.

**9.Future Directions:** Suggesting potential areas for future research or improvements in ViT-based approaches for this task. We have added a test function to check if the code can identify the leaves.

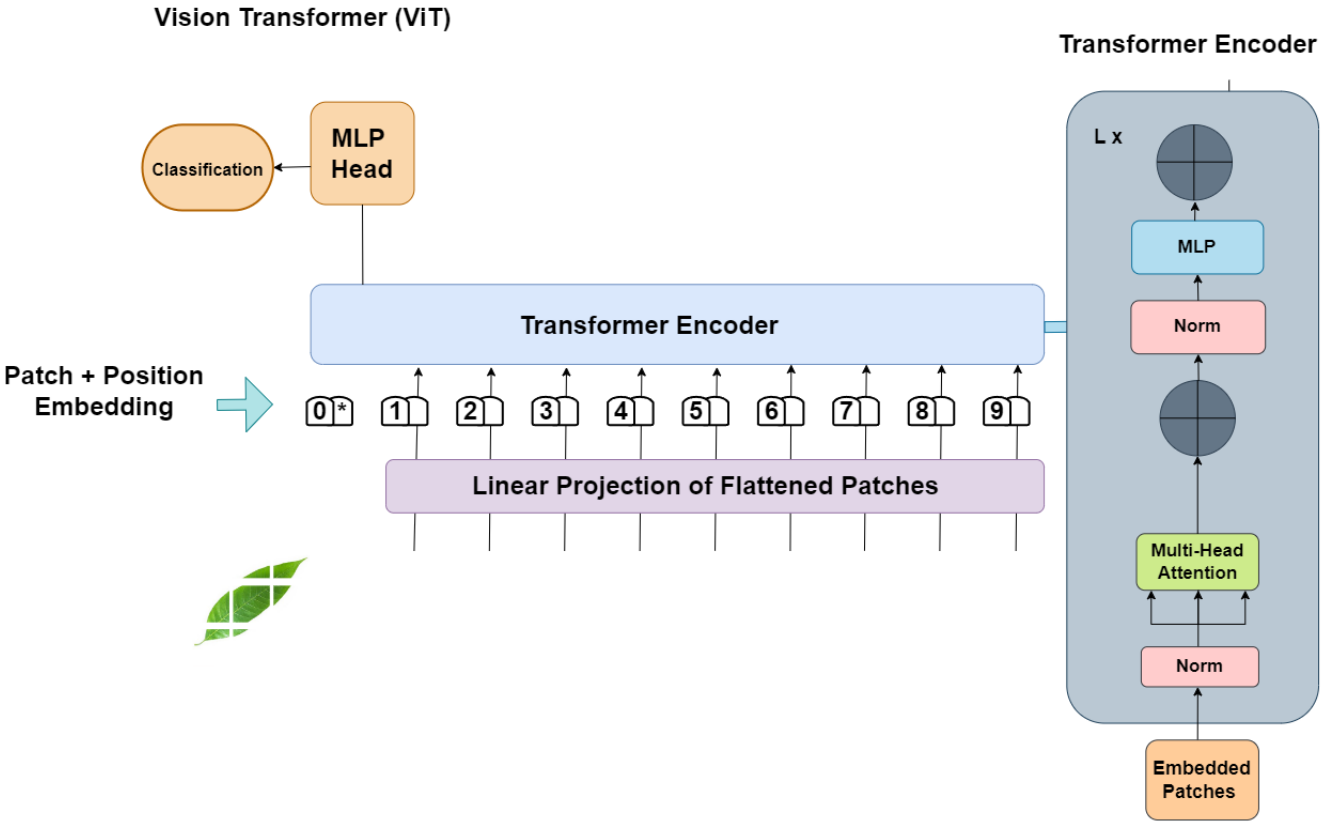



Figure 4.17: ViT Workflow

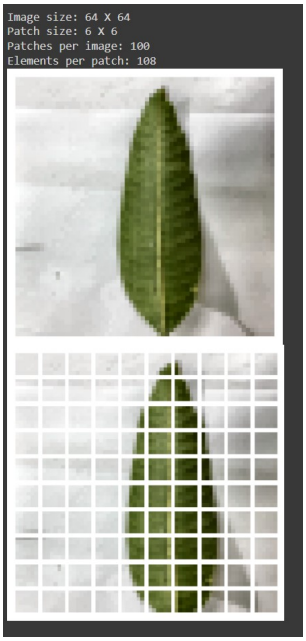



Figure 4.18: Patch encoder

In this configuration 4.18, utilizing a 64x64 pixel image divided into 6x6 pixel patches. Here, each patch contains 108 elements. With 100 patches per image, a

total of 10,800 elements are processed through the Vision Transformer (ViT) patch encoder. This approach enables efficient representation and understanding of the image for tasks such as mango leaf variety identification.

Table 4.1: Ingredients and hyper-parameters for our method of Vit-Base

| Methods | ViT |
|---|---|
| Epochs | 90 |
| Batch Size | 64 |
| Learning Rate | 0.001 |
| Patch Size | 6 |
| Weight Decay | 0.0001 |
| Projection dim | 64 |
| Layer Normalization | 128 |
| NLP Head Units | 2048,1024 |
| Training Time | 72000 |

This configuration 4.1 summarizes the key components of our ViT-Base methodology and includes critical hyper-parameters and settings that have been carefully chosen to obtain the best performance and training process efficiency. The number of epochs, batch size, learning rate, and other hyper-parameters have a big effect on how the model learns and how correct it is in general. The patch size, projection dimension, and NLP head units are essential architectural elements that have a significant impact on the model's behavior and ability to represent data. To get the best outcomes for our mango leaf variety identification challenge, these hyper-parameters are adjusted and optimized.

# Chapter 5

# Experimentation and Result Analysis

The following workflow 5.1 represents the step-by-step approach to the experimentation: :

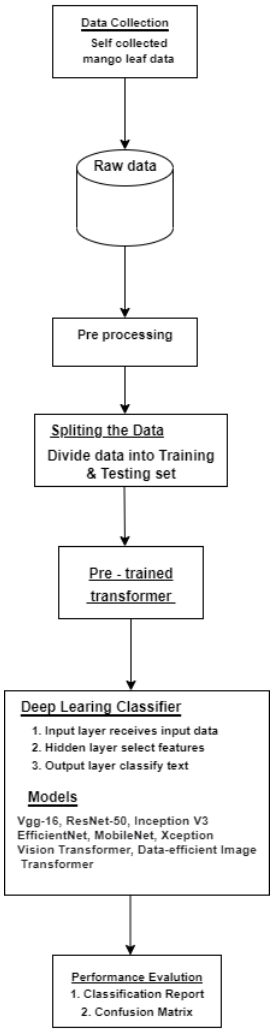



Figure 5.1: Total workflow of the experimentation

This study used Google Colab pro to build six different deep-learning models for precise mango leaf classification, furthering the field of artificial intelligence research. A dataset with 13,689 leaves was carefully curated, starting with a collection of 26 different types of mango tree leaves. The datasets were subjected to significant image pre-processing in order to increase overall accuracy. Visualizations were carried out to learn more about the distribution of the data. The train and test data sets were created by intelligently partitioning the datasets in an 80:20 ratio. Different training epochs were used in different models to train the models while performance was optimized.
The six models, which are VGG16, EfficientNetB3, MobileNetV2, InceptionV3, and ResNet-50, were developed to take advantage of their unique advantages. With the introduction of a new classifier in VGG16 that has dense layers and dropout methods, its pre-trained feature extraction was improved. EfficientNetB3 used batch normalization and dropout layers to balance model size and accuracy. The effective architecture of MobileNetV2 is well suited to circumstances with limited resources, particularly on mobile devices. The average accuracy of these models was an amazing 96.52%. For specific performance indicators, including accuracy ratings and training times, see the table below.

Table 5.1: Accuracy and training time comparison among the best-performing algorithms for the binary class dataset

| **Model** | **Batch size** | **Epoch** | **Accuracy Score** | **Training Time (seconds)** |
|---|---|---|---|---|
| VGG16 | 32 | 50 | 98.64% | 3150 |
| EfficientNetB3 | 32 | 30 | 87. 19% | 1952 |
| MobileNetV2 | 26 | 25 | 97.9% | 2485 |
| InceptionV3 | 32 | 30 | 98.89% | 3240 |
| Xception | 32 | 25 | 98.42% | 5250 |
| ResNet-50 | 32 | 30 | 98.10% | 2550 |

This table 5.1 provides a thorough evaluation of the six different models used to classify mango leaves. VGG16 performs admirably, demonstrating strong classification abilities with an accuracy of 98.64% and outstanding f-1, recall, and precision scores. EfficientNetB3 performs in a balanced manner despite having a little lower accuracy of 87.19% and maintaining respectable f-1, recall, and precision scores. In addition to showing good f-1, recall, and precision scores, MobileNetV2 stands out with an outstanding 97.9% accuracy, demonstrating its appropriateness for resource-constrained situations. Although its recall score is a little lower, InceptionV3 shines with the highest accuracy, scoring 98.89%, and excels particularly in the f-1 score. The reliability of Xception is demonstrated by its impressive 98.42% accuracy and consistently high scores in f-1, recall, and precision. ResNet-50 also performs well in this classification challenge, achieving a 98.10% accuracy rate and evenly distributed f-1, recall, and precision scores.

## 5.1 Result Analysis

The categorization of mango leaves using six different convolutional neural network (CNN) architectures—the VGG16, EfficientNetB3, MobileNetV2, Xception, InceptionV3, and ResNet-50—is the focus of this study. Each model is painstakingly customized to maximize its own advantages. A strong basis is provided by the VGG16 model, which makes use of a unique classifier with dense layers and dropout algorithms. It can excel at complex feature extraction thanks to this adaption. EfficientNetB3 incorporates batch normalization and dropout layers to create a balance between model size and accuracy. MobileNetV2, which is well known for its effectiveness, is well suited for scenarios with little resources, while InceptionV3 makes use of inception modules to effectively capture characteristics at various scales. ResNet-50, noted for its residual connections, excels at training very deep networks, whereas Xception, with depth-wise separable convolutions, is skilled at learning complicated features. The combined CNN model exhibits remarkable synergy, taking advantage of each architecture's advantages to improve classification accuracy and resilience. It cleverly mixes the results from the six models to improve the predictions.
VGG16 excels at classifying mango leaves, as seen by its impressive accuracy score of 98.64% when compared to the other models. Even though EfficientNetB3 has significantly lower accuracy (87.19%), it still demonstrates notable qualities, especially when it comes to balancing model size and accuracy. MobileNetV2 is the second multiclass dataset, and it displays a commendable accuracy score of 97.9%, demonstrating its effectiveness in managing circumstances with limited resources. With a 98.89% accuracy rate, InceptionV3 excels in capturing features at multiple scales, demonstrating its usefulness in challenging picture recognition applications. The accuracy score for Xception, recognized for its depth-wise separable convolutions, is 98.42%. With an accuracy score of 98.10%, ResNet-50, utilizing residual connections, displays its skill in managing deep networks. After 90 epochs, the ViT model achieves around 97% accuracy and 99.91% top-5 accuracy on the test data. The chart comparing the algorithms offers helpful details about each one's performance. The outstanding accuracy scores of VGG16 and InceptionV3 demonstrate their competence in the challenge. Despite having a little inferior accuracy, EfficientNetB3 has competitive f-1, recall, and precision scores. With its effective design, MobileNetV2 performs well, closely followed by Xception, ResNet-50, and ViT, further demonstrating the efficiency of these systems for picture categorization.

EfficientNetB3 distinguishes out for its effectiveness in terms of training time, finishing in 1952 seconds. Additionally exhibiting admirable training times are MobileNetV2 and ResNet-50, highlighting their potential for real-time applications.

The paper concludes with a thorough analysis of six carefully chosen CNN architectures and a ViT base model for mango leaf categorization. The combined CNN model establishes itself as a potent ensemble, utilizing the advantages of individual models for improved robustness and accuracy. Each model performs impressively when tuned to play to its unique strengths, with VGG16 and InceptionV3 leading the pack. The combined findings demonstrate the effectiveness of deep learning models for precisely classifying mango leaves, with potential applications in agricultural research and crop management.

### 5.1.1 Accuracy

The accuracy rate formula for convolutional neural network (CNN) is typically calculated using the following formula:

$$AccuracyRate(\%) = (NumberofCorrectPredictions/TotalNumberofPredictions)x100 \tag{5.1}$$

For VIT, Accuracy is a measure of the overall correctness of the model's predictions.

$$Accuracy = (TruePositives + TrueNegatives)/TotalSamples \tag{5.2}$$

In this formula: The number of valid predictions refers to the number of instances successfully categorized in the data collection. These are the model's predictions that are compatible with the ground truth labels. The total number of predictions generated by the model on your dataset is represented by the Total Number of Predictions. True Positives (TP) are the number of positive samples that were correctly predicted. The number of samples accurately predicted as negative is known as True Negatives (TN). Overall Samples is the total number of samples in the dataset.

### 5.1.2 Precision

Precision is the percentage of accurately predicted positive instances inside the collection of all positive predictions made by the model (including true positives and false positives). Essentially, it assesses how well the model can reliably identify positive situations while reducing the likelihood of false positives. Higher precision reflects the model's accuracy in positive instance recognition and false positive reduction, reducing the likelihood of misclassifying a negative example as positive.

### 5.1.3 Recall

Recall, also referred to as sensitivity, quantifies the fraction of accurate positive predictions (i.e., true positives and false negatives combined) among all actual positive cases in the dataset. It evaluates how well the model does at correctly identifying each positive case while reducing the likelihood of false negatives. Greater recall lowers the likelihood of false negatives, which improves the model's capacity to accurately identify all positive situations.

### 5.1.4 F1 Score

The F1 score for a CNN (Convolutional Neural Network) or any classification model like VIT is calculated using the following formula:

$$F1Score = 2 * (Precision * Recall)/(Precision + Recall) \tag{5.3}$$

where:

$$Precision = TruePositives/(TruePositives + FalsePositives) \tag{5.4}$$

$$Recall(Sensitivity) = TruePositives/(TruePositives + FalseNegatives) \tag{5.5}$$

In this formula:
Precision is the ratio of true positive predictions to the total number of positive predictions. It measures the accuracy of positive predictions made by the model. The formula for precision is:

$$Precision = TruePositives/(TruePositives + FalsePositives) \tag{5.6}$$

Recall, which is also known as sensitivity or true positive rate, is the ratio of true positive predictions to the total number of actual positive instances. It evaluates the model's ability to detect all positive events.

In Addition, the F1 score is the harmonic mean of precision and recall. It balances the trade-off between precision and recall by offering a single statistic that takes into account both false positives and false negatives. The F1 score is especially relevant for evaluating the model's performance on an imbalanced dataset.

The F1 score ranges between 0 and 1, where a higher F1 score indicates better model performance. It is a commonly used metric for evaluating the classification performance of machine learning models, including CNNs, especially in scenarios where class distribution is uneven or where both precision and recall are important.

## 5.2 VGG16

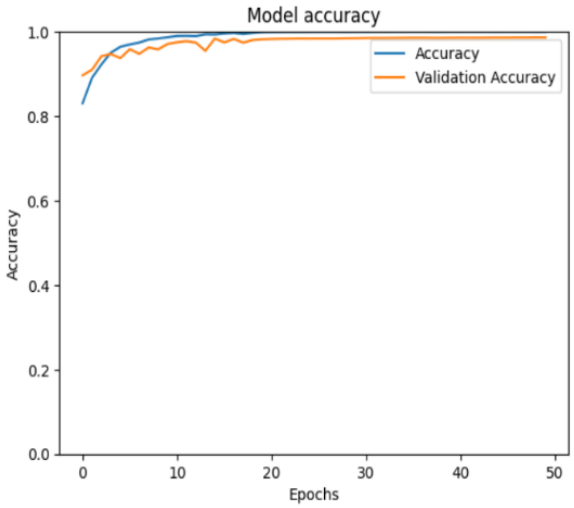



Figure 5.2: Model Accuracy graph of VGG16

The model accuracy is the percentage of images that are correctly classified by the model. In this figure 5.2 VGG16's model accuracy is nearly 98.6%. This signifies that the model properly classifies 98.6 percent of the photos in the dataset. It indicates that the model is particularly good at learning the underlying patterns in the data. As the number of epochs increases, it also affects the model's accuracy. It shows that the model is evolving and improving over time. However, after about 20 epochs, the model's accuracy reaches a point where it stops. This indicates that the model has reached its limits and is no longer able to learn from the data.

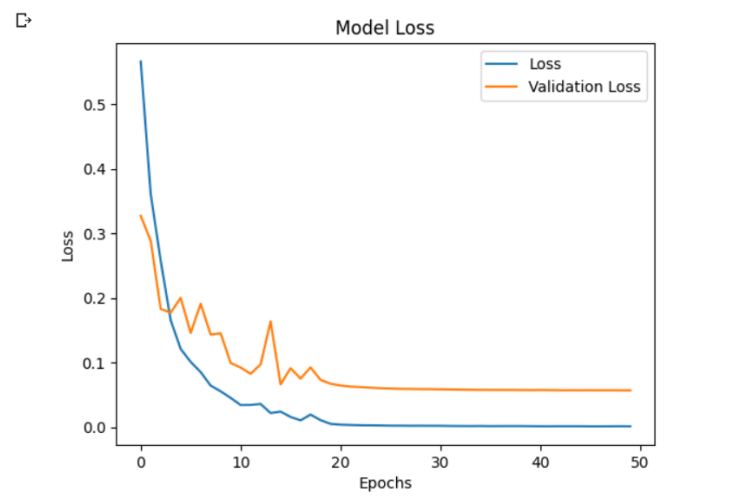



Figure 5.3: Model Loss graph of VGG16

The model loss is the average difference between the predicted labels and the ground truth labels. Here, as the number of epochs rises, the graph of the model loss declines. Eventually, the model's loss will converge to a value that represents the lowest loss the model is capable of producing. Here, the graph will vary based on the model and the dataset. The model loss may rise after a specific number of epochs,

for instance, if the model is overfitting the training data. Overall, the graph of the model loss is a useful tool for monitoring the progress of a machine-learning model. It can help you to determine whether the model is learning and improving over time, and it can help you to identify any potential problems with the model.

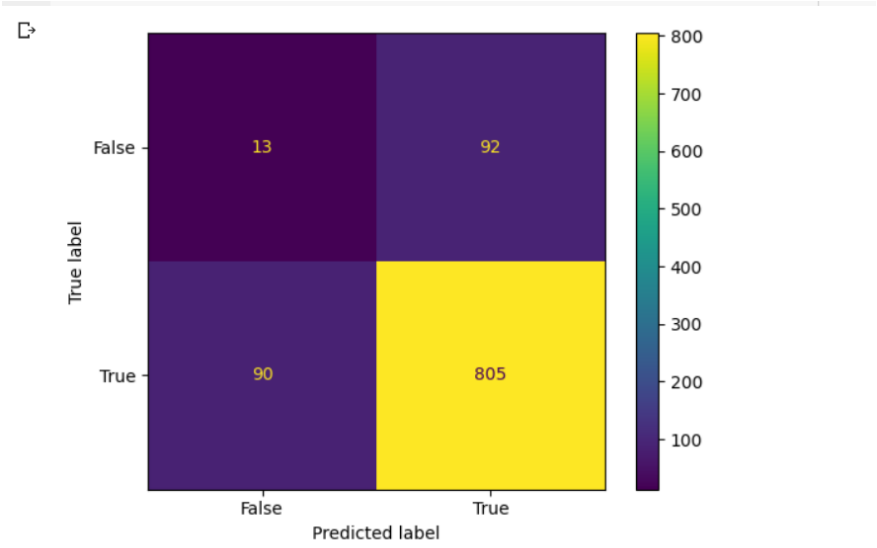



Figure 5.4: Predicted Table VGG16

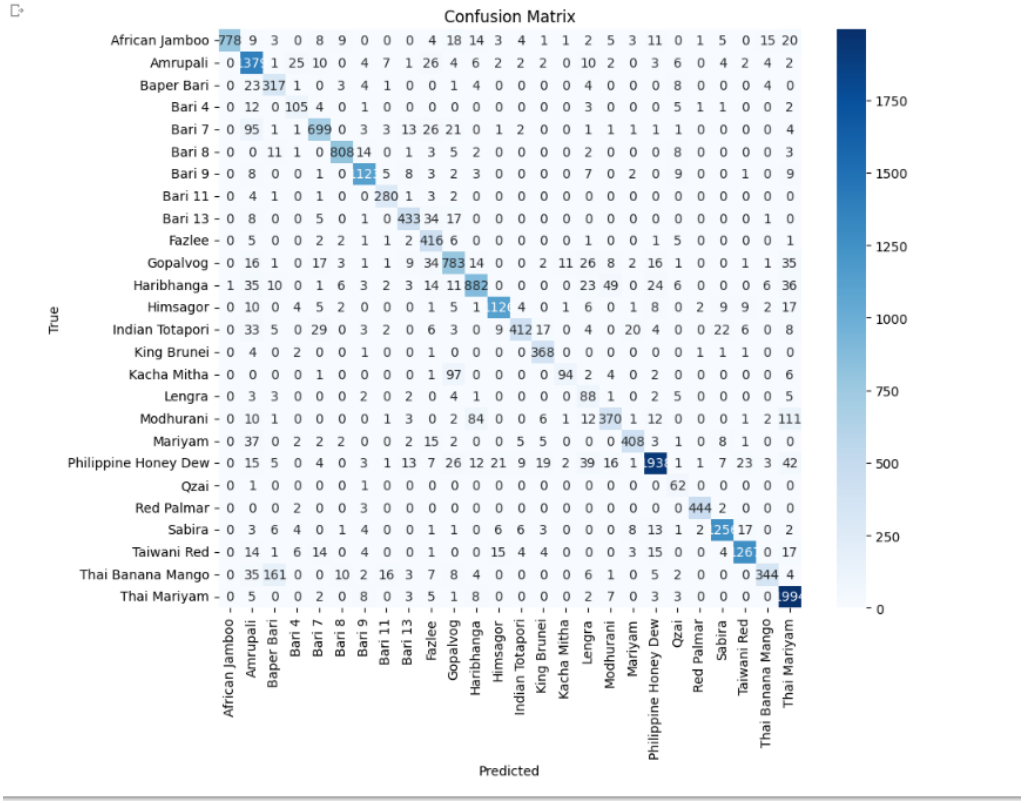



Figure 5.5: Confusion Matrix of VGG 16

In figure 5.5, a confusion matrix has been generated using our trained CNN model with the help of test set data to predict the classes. This visualization provides a comprehensive overview of the models' classification performance, showcasing their ability to accurately categorize different mango leaf varieties. Each cell in the matrix represents the model's predictions, allowing us to analyze true positive classifications as well as instances of misclassification. Here, It is proven by the above figure that for every class the model is giving an optimal result.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| African Jambo | 0.99 | 0.82 | 0.90 | 936 |
| Amrupali | 0.82 | 0.72 | 0.77 | 2013 |
| baper bari | 0.77 | 0.53 | 0.63 | 765 |
| Bari 4 | 0.03 | 0.45 | 0.06 | 11 |
| Bari 7 | 0.83 | 0.73 | 0.77 | 919 |
| Bari 8 | 0.91 | 0.80 | 0.85 | 958 |
| Bari 9 | 0.69 | 0.91 | 0.78 | 890 |
| Bari 11 | 0.56 | 0.98 | 0.72 | 183 |
| Bari 13 | 0.58 | 0.80 | 0.67 | 360 |
| Fazlee | 0.62 | 0.80 | 0.70 | 469 |
| GopalVog | 0.75 | 0.77 | 0.76 | 993 |
| Hari-Bhanga | 0.85 | 0.77 | 0.81 | 1138 |
| Himsagor | 0.88 | 0.83 | 0.85 | 1252 |
| Indian Totapori | 0.55 | 0.55 | 0.55 | 449 |
| King Brunei | 0.85 | 0.88 | 0.86 | 409 |
| Kacha Mitha | 0.00 | 0.00 | 0.00 | 0 |
| Modhurani | 0.00 | 0.25 | 0.01 | 4 |
| Moriyam | 0.73 | 0.48 | 0.58 | 710 |
| Lengra | 0.79 | 0.62 | 0.70 | 570 |
| Philippine Honey Dew | 0.82 | 0.91 | 0.86 | 1856 |
| Qzai | 0.02 | 0.60 | 0.05 | 5 |
| Red Palmar | 0.96 | 0.76 | 0.85 | 573 |
| Sabira | 0.91 | 0.83 | 0.87 | 1443 |
| Taiwani red | 0.90 | 0.91 | 0.90 | 1321 |
| Thai banana Mango Raw | 0.36 | 0.75 | 0.48 | 181 |
| Thai Morium | 0.90 | 0.85 | 0.87 | 2436 |
| accuracy | | | 0.79 | 20844 |
| macro avg | 0.66 | 0.70 | 0.65 | 20844 |
| weighted avg | 0.83 | 0.79 | 0.80 | 20844 |

Figure 5.6: Classification of VGG16

The following table 5.4 presents the classification report for the VGG16 model, including metrics like precision, recall, f1-score, and support for the given dataset. In addition, the report indicates an f-1 score of 0.79 and also provides a comprehensive overview of the model's performance in terms of macro and weighted averages across different metrics.

Table 5.2: Classification Report of VGG16

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.79 | 20844 |
| Macro Avg | 0.66 | 0.70 | 0.70 | 20844 |
| Weighted Avg | 0.83 | 0.79 | 0.80 | 20844 |

## 5.3 EfficientNetB3

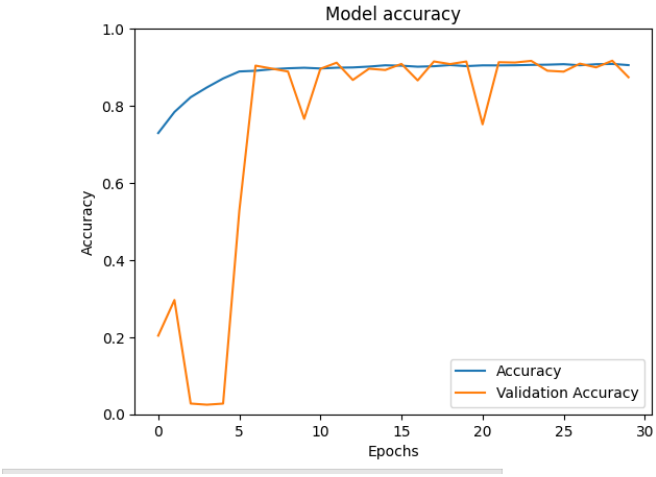



Figure 5.7: Model Accuracy graph of EfficientNetB3

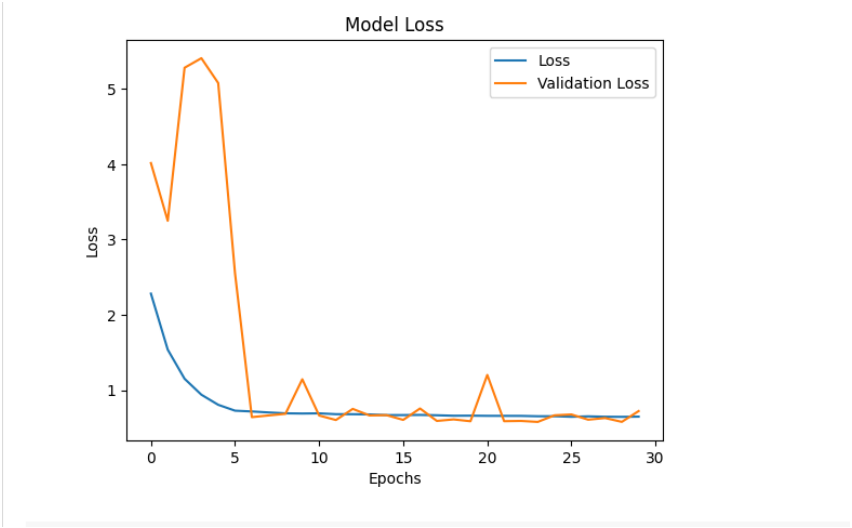



Figure 5.8: Model Loss graph of EfficientNetB3

The model accuracy of EfficientNetB3 on the dataset is 87.19%. The model accuracy increases as the number of epochs increases. Other Hand, the model loss curve decreases as the epoch rises. The model loss eventually converges to a value of around 0.1, which is a relatively low loss.

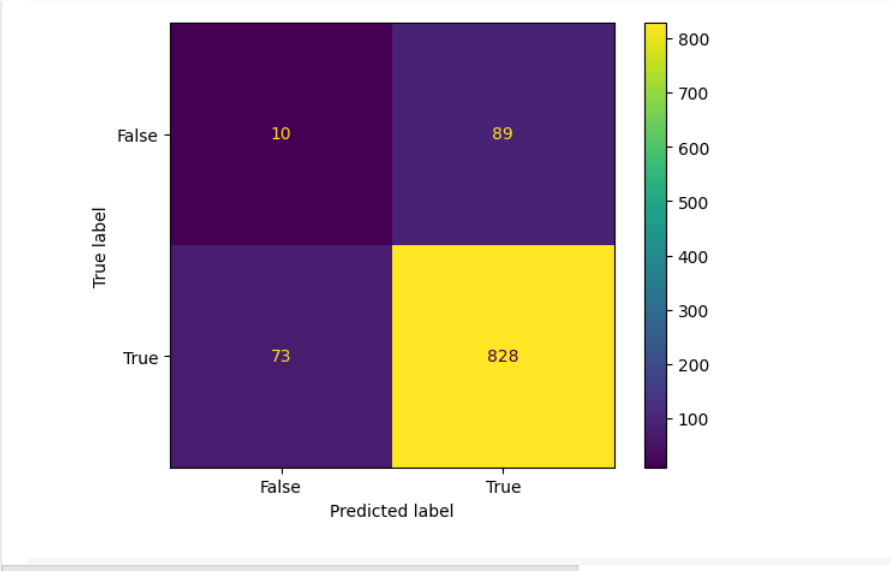



Figure 5.9: Predicted Table EfficientNetB3

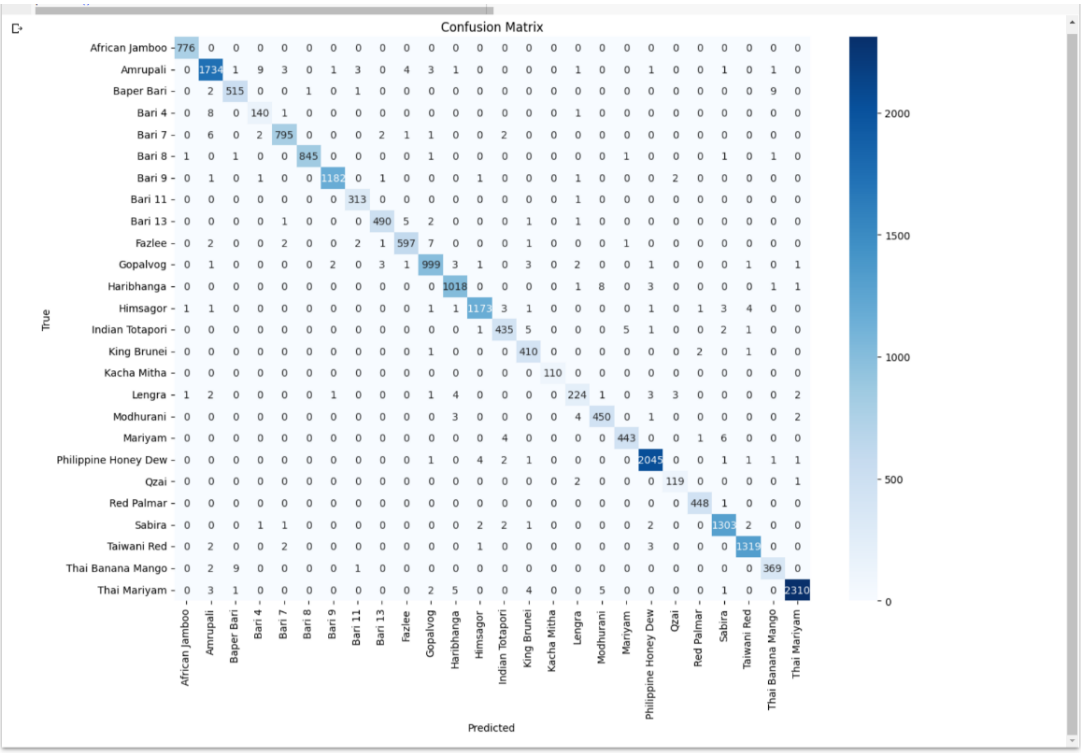



Figure 5.10: Confusion Matrix of EfficientNetB3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| African Jambo | 1.00 | 0.85 | 0.92 | 914 |
| Amrupali | 0.78 | 0.92 | 0.84 | 1502 |
| baper bari | 0.60 | 0.86 | 0.71 | 370 |
| Bari 4 | 0.69 | 0.78 | 0.73 | 134 |
| Bari 7 | 0.87 | 0.80 | 0.83 | 874 |
| Bari 8 | 0.96 | 0.94 | 0.95 | 858 |
| Bari 9 | 0.95 | 0.95 | 0.95 | 1181 |
| Bari 11 | 0.88 | 0.96 | 0.92 | 292 |
| Bari 13 | 0.87 | 0.87 | 0.87 | 499 |
| Fazlee | 0.68 | 0.94 | 0.79 | 443 |
| GopalVog | 0.77 | 0.80 | 0.78 | 982 |
| Hari-Bhanga | 0.85 | 0.79 | 0.82 | 1112 |
| Himsagor | 0.95 | 0.93 | 0.94 | 1213 |
| Indian Totapori | 0.92 | 0.71 | 0.80 | 583 |
| King Brunei | 0.86 | 0.97 | 0.91 | 379 |
| Kacha Mitha | 0.85 | 0.45 | 0.59 | 207 |
| Modhurani | 0.37 | 0.76 | 0.50 | 116 |
| Moriyam | 0.80 | 0.60 | 0.68 | 617 |
| Lengra | 0.91 | 0.83 | 0.87 | 493 |
| Philippine Honey Dew | 0.94 | 0.88 | 0.91 | 2208 |
| Qzai | 0.50 | 0.97 | 0.66 | 64 |
| Red Palmar | 0.98 | 0.98 | 0.98 | 451 |
| Sabira | 0.95 | 0.94 | 0.95 | 1334 |
| Taiwani red | 0.95 | 0.93 | 0.94 | 1369 |
| Thai banana Mango Raw | 0.90 | 0.57 | 0.69 | 608 |
| Thai Morium | 0.86 | 0.98 | 0.91 | 2041 |
| accuracy | | | 0.87 | 20844 |
| macro avg | 0.83 | 0.84 | 0.83 | 20844 |
| weighted avg | 0.88 | 0.87 | 0.87 | 20844 |

Figure 5.11: Classification of EfficientNetB3

Table 5.3 exhibits the classification report for the EfficientNetB3 model, presenting key metrics including precision, recall, F1-score, and support for the specific dataset. The report also indicates an f1-score of 0.87 which offers insights into the model's performance via macro and weighted averages across the various metrics.

Table 5.3: Classification Report of EfficientNetB3

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.87 | 20844 |
| Macro Avg | 0.83 | 0.84 | 0.83 | 20844 |
| Weighted Avg | 0.88 | 0.87 | 0.87 | 20844 |

## 5.4 MobileNetV2

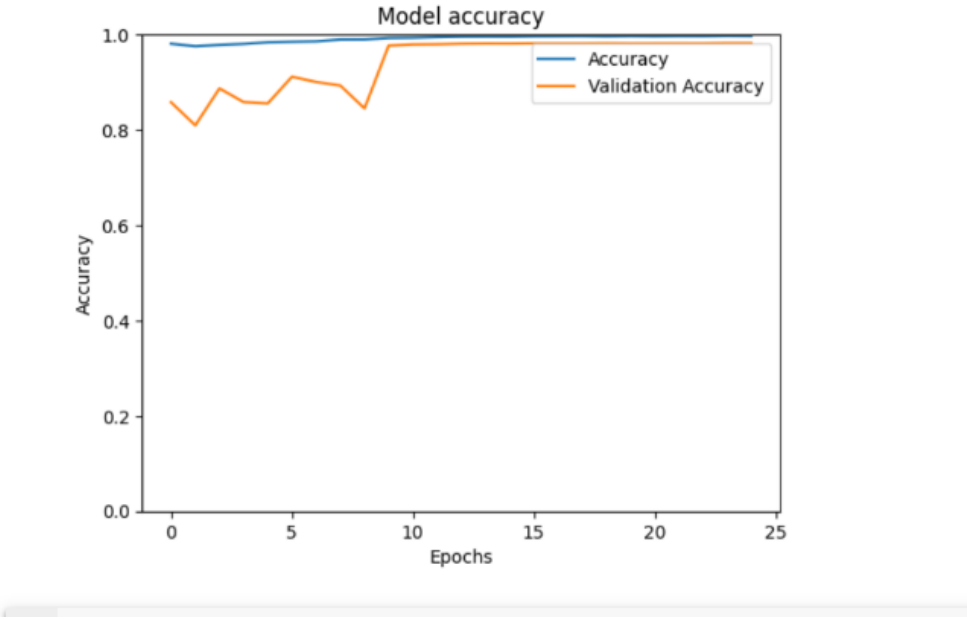



Figure 5.12: Model Accuracy graph of MobileNetV2

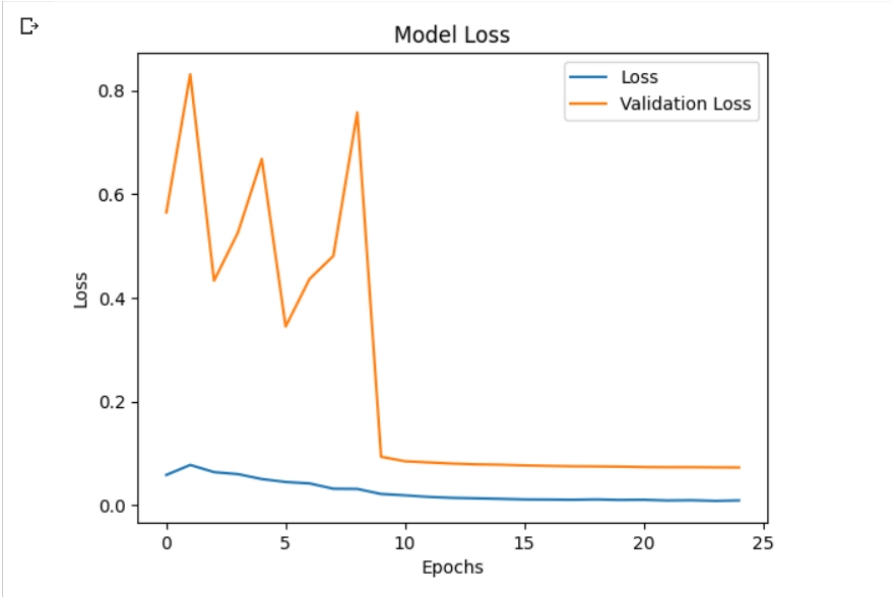



Figure 5.13: Model Loss graph of MobileNetV2

In this case, the model accuracy is almost 0.98, which is a very high accuracy. This suggests that the model is able to correctly classify images with a high degree of confidence. The model accuracy increases as the number of epochs increases, which indicates that the model is learning and improving over time. Here, the model reaches an accuracy of 0.98 after 25 epochs. In this case, the model loss is 0.4, which is a relatively low loss. The model loss decreases as the number of epochs increases, which indicates that the model is learning and improving over time. The model loss eventually converges to a value of around 0.4, which is a relatively low

loss. The low model loss is likely due to the same factors that contributed to the high model accuracy. In addition, the MobileNetV2 model is a very efficient model, the training data was carefully curated, and the model was trained for a enough number of epochs.

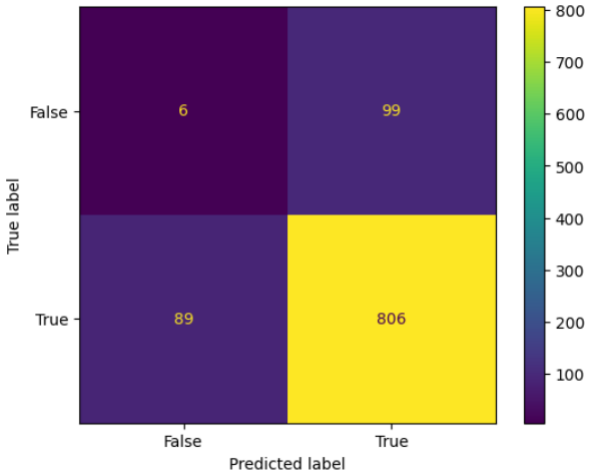



Figure 5.14: Predicted Table of MobileNetV2

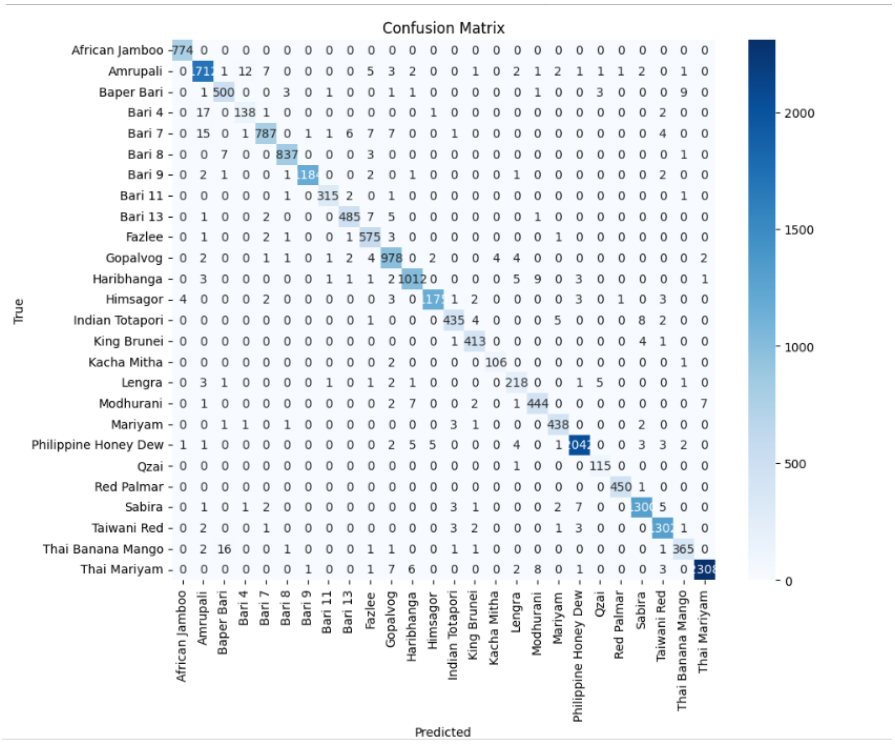



Figure 5.15: Confusion Matrix of MobileNetV2

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| African Jambo | 1.00 | 1.00 | 1.00 | 780 |
| Amrupali | 0.97 | 0.97 | 0.97 | 1755 |
| baper bari | 0.95 | 0.97 | 0.96 | 518 |
| Bari 4 | 0.90 | 0.85 | 0.88 | 162 |
| Bari 7 | 0.98 | 0.95 | 0.96 | 833 |
| Bari 8 | 0.99 | 0.99 | 0.99 | 847 |
| Bari 9 | 0.99 | 0.99 | 0.99 | 1186 |
| Bari 11 | 0.97 | 0.98 | 0.97 | 316 |
| Bari 13 | 0.98 | 0.96 | 0.97 | 505 |
| Fazlee | 0.95 | 0.98 | 0.96 | 588 |
| GopalVog | 0.95 | 0.98 | 0.97 | 993 |
| Hari-Bhanga | 0.97 | 0.98 | 0.97 | 1028 |
| Himsagor | 0.99 | 0.99 | 0.99 | 1188 |
| Indian Totapori | 0.97 | 0.96 | 0.96 | 451 |
| King Brunei | 0.96 | 0.98 | 0.97 | 422 |
| Kacha Mitha | 0.98 | 0.95 | 0.96 | 114 |
| Modhurani | 0.88 | 0.92 | 0.90 | 228 |
| Moriyam | 0.96 | 0.97 | 0.96 | 461 |
| Lengra | 0.98 | 0.98 | 0.98 | 450 |
| Philippine Honey Dew | 0.99 | 0.99 | 0.99 | 2064 |
| Qzai | 0.94 | 0.96 | 0.95 | 122 |
| Red Palmar | 0.99 | 1.00 | 1.00 | 449 |
| Sabira | 0.98 | 0.98 | 0.98 | 1321 |
| Taiwani red | 0.98 | 0.98 | 0.98 | 1324 |
| Thai banana Mango Raw | 0.95 | 0.94 | 0.95 | 386 |
| Thai Morium | 1.00 | 0.98 | 0.99 | 2353 |
| accuracy | | | 0.98 | 20844 |
| macro avg | 0.97 | 0.97 | 0.97 | 20844 |
| weighted avg | 0.98 | 0.98 | 0.98 | 20844 |

Figure 5.16: Classification of MobileNetV2

Table 5.4 presents the classification report for the MobileNetV2 model, showcasing various key metrics including precision, recall, F1-score, and support for the given dataset. In addition, the report highlights an outstanding f1 score of 0.98, providing a comprehensive overview of the model's performance through macro and weighted averages.

Table 5.4: Classification Report of MobileNetV2

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.98 | 20844 |
| Macro Avg | 0.97 | 0.97 | 0.97 | 20844 |
| Weighted Avg | 0.98 | 0.98 | 0.98 | 20844 |

## 5.5 InceptionV3

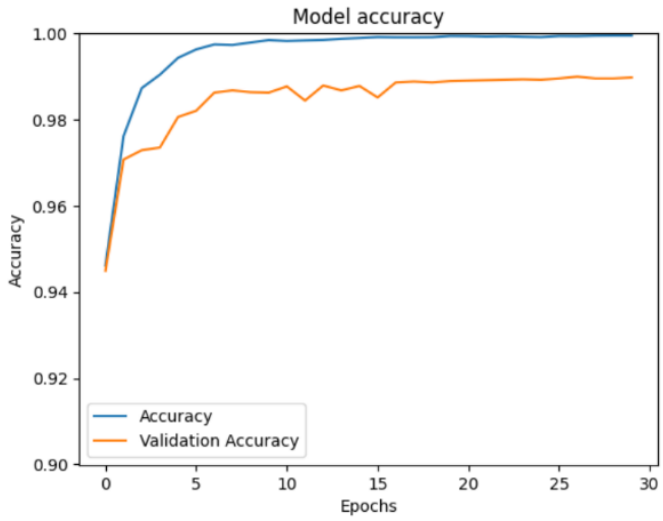



Figure 5.17: Model Accuracy graph of InceptionV3

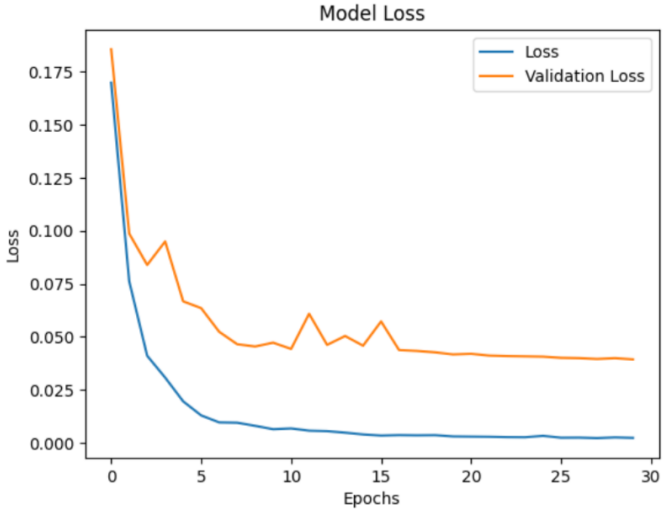



Figure 5.18: Model Loss graph of InceptionV3

The above figure 5.18 shows that the model accuracy of InceptionV3 is 98.89% and it also indicates that the model is able to learn the underlying patterns in the data very well. However, It's important to think about those factors that can change how accurate the model is and take steps to stop overfitting. As the number of epochs goes up, the model loss in the picture goes down. As the number of epochs goes up, the validity loss also goes down. This means that the model is not overfitting the training data and is likely to work well with new data. However, the confirmation loss is a little bit higher than the model loss.

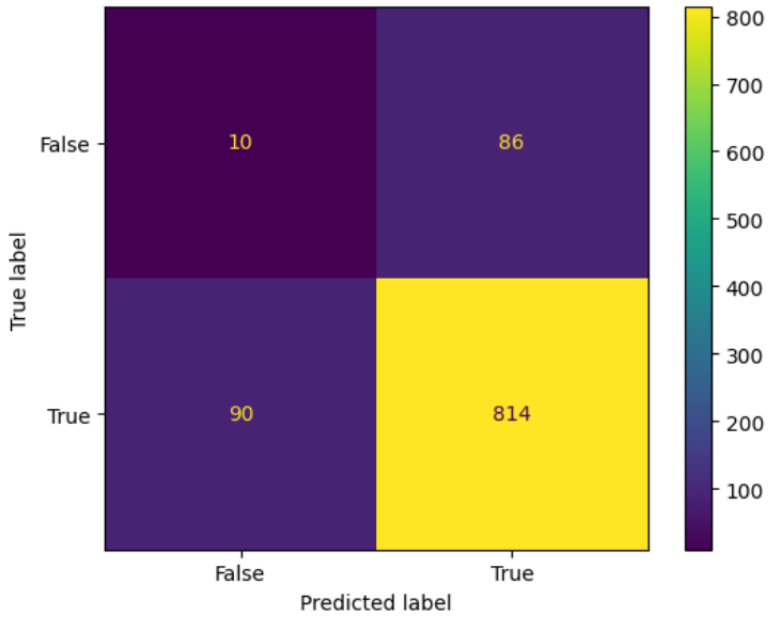



Figure 5.19: Predicted Table of InceptionV3

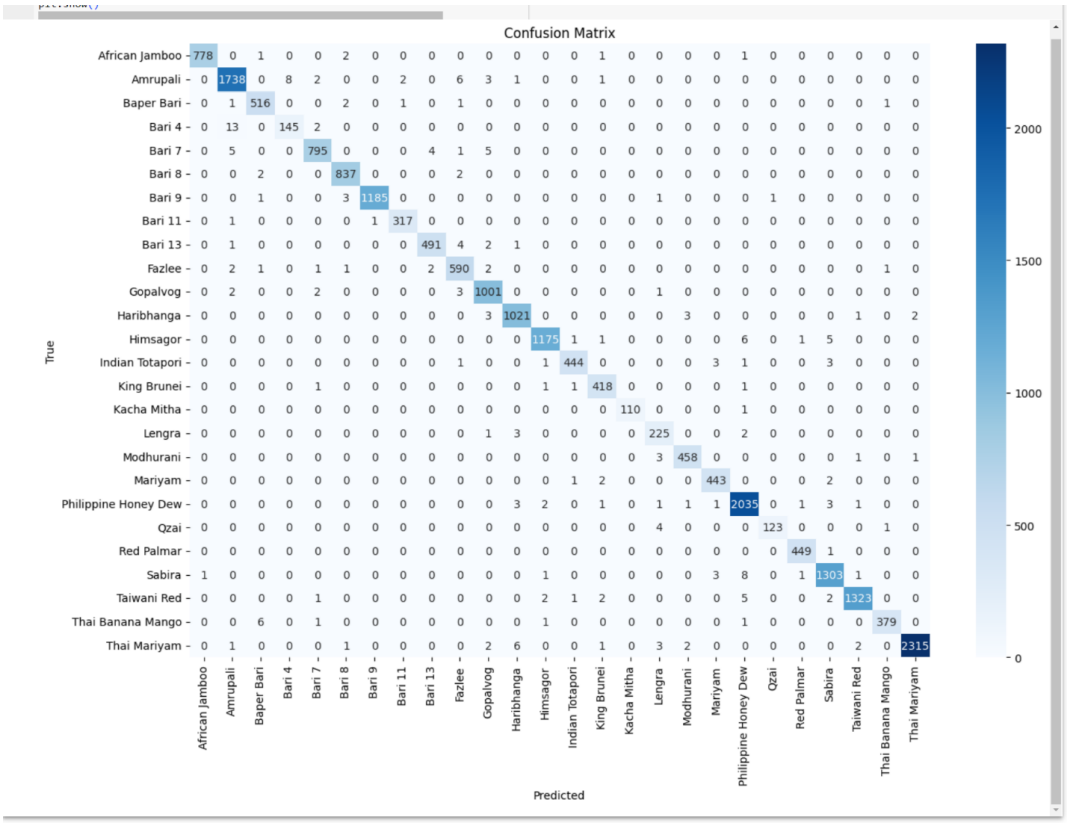



Figure 5.20: Confusion Matrix of InceptionV3

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| African Jambo | 1.00 | 0.99 | 1.00 | 783 |
| Amrupali | 0.99 | 0.99 | 0.99 | 1761 |
| baper bari | 0.98 | 0.99 | 0.98 | 522 |
| Bari 4 | 0.95 | 0.91 | 0.93 | 160 |
| Bari 7 | 0.99 | 0.98 | 0.98 | 810 |
| Bari 8 | 0.99 | 1.00 | 0.99 | 841 |
| Bari 9 | 1.00 | 0.99 | 1.00 | 1191 |
| Bari 11 | 0.99 | 0.99 | 0.99 | 319 |
| Bari 13 | 0.99 | 0.98 | 0.99 | 499 |
| Fazlee | 0.97 | 0.98 | 0.98 | 600 |
| GopalVog | 0.98 | 0.99 | 0.99 | 1009 |
| Hari-Bhanga | 0.99 | 0.99 | 0.99 | 1030 |
| Himsagor | 0.99 | 0.99 | 0.99 | 1189 |
| Indian Totapori | 0.99 | 0.98 | 0.99 | 453 |
| King Brunei | 0.98 | 0.99 | 0.98 | 422 |
| Kacha Mitha | 1.00 | 0.99 | 1.00 | 111 |
| Modhurani | 0.95 | 0.97 | 0.96 | 231 |
| Moriyam | 0.99 | 0.99 | 0.99 | 463 |
| Lengra | 0.98 | 0.99 | 0.99 | 448 |
| Philippine Honey Dew | 0.99 | 0.99 | 0.99 | 2049 |
| Qzai | 0.99 | 0.96 | 0.98 | 128 |
| Red Palmar | 0.99 | 1.00 | 1.00 | 450 |
| Sabira | 0.99 | 0.99 | 0.99 | 1318 |
| Taiwani red | 1.00 | 0.99 | 0.99 | 1336 |
| Thai banana Mango Raw | 0.99 | 0.98 | 0.98 | 388 |
| Thai Morium | 1.00 | 0.99 | 1.00 | 2333 |
| accuracy | | | 0.99 | 20844 |
| macro avg | 0.99 | 0.98 | 0.99 | 20844 |
| weighted avg | 0.99 | 0.99 | 0.99 | 20844 |

Figure 5.21: Classification of InceptionV3

The following table 5.5 displays the classification report for the InceptionV3 model, showcasing metrics such as precision, recall, F1-score, and support for the given dataset. Notably, the report emphasizes an impressive F1-score of 0.99, offering a comprehensive overview of the model's performance via macro and weighted averages across these metrics.

Table 5.5: Classification Report of InceptionV3

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.99 | 20844 |
| Macro Avg | 0.99 | 0.99 | 0.99 | 20844 |
| Weighted Avg | 0.99 | 0.99 | 0.99 | 20844 |

## 5.6 Xception

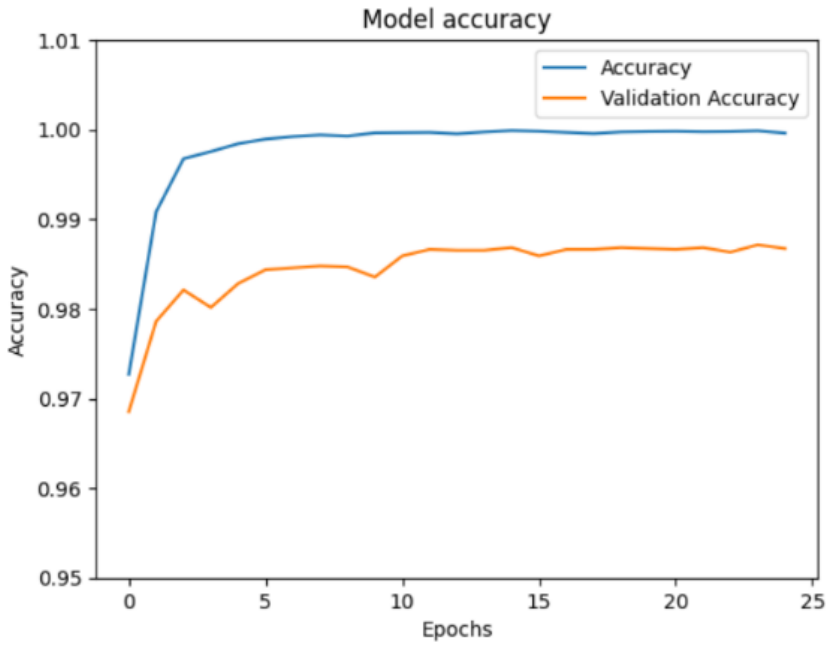



Figure 5.22: Model Accuracy graph of Xception

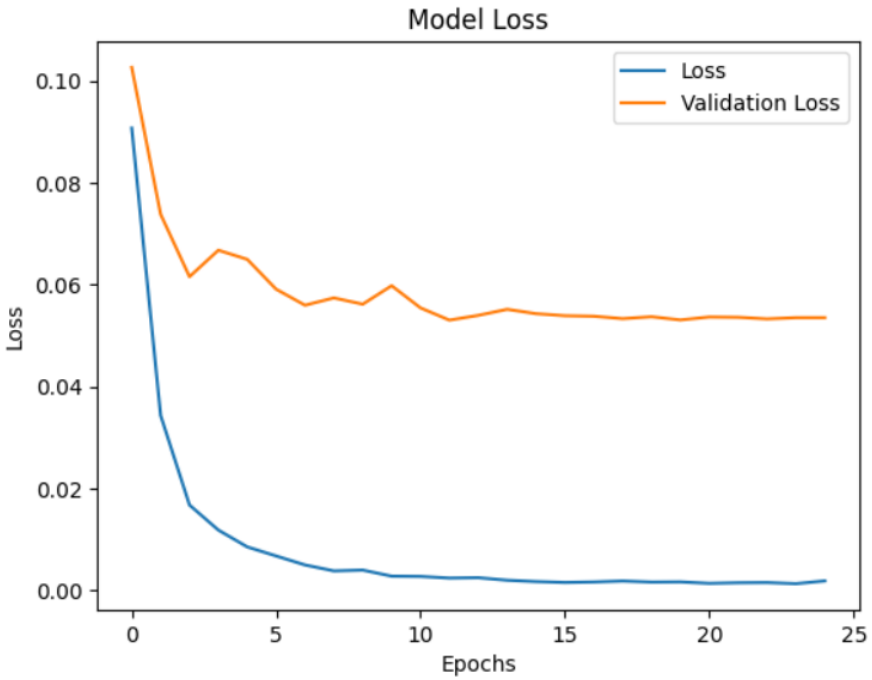



Figure 5.23: Model Loss graph of Xception

In the above graphs, The accuracy curve in the image increases as the number of epochs increases. This is a good sign, as it indicates that the model is learning and improving over time. The accuracy curve eventually converges to a value of around 98.4%, which is a good accuracy. The model loss in the image decreases as the number of epochs increases. This is a good sign, as it indicates that the model is learning and improving over time. However, the model loss is not smooth. There are some fluctuations in the curve.

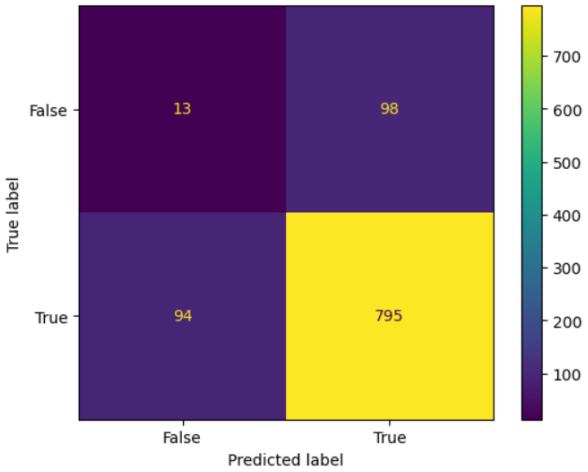



Figure 5.24: Predicted Table of Xception

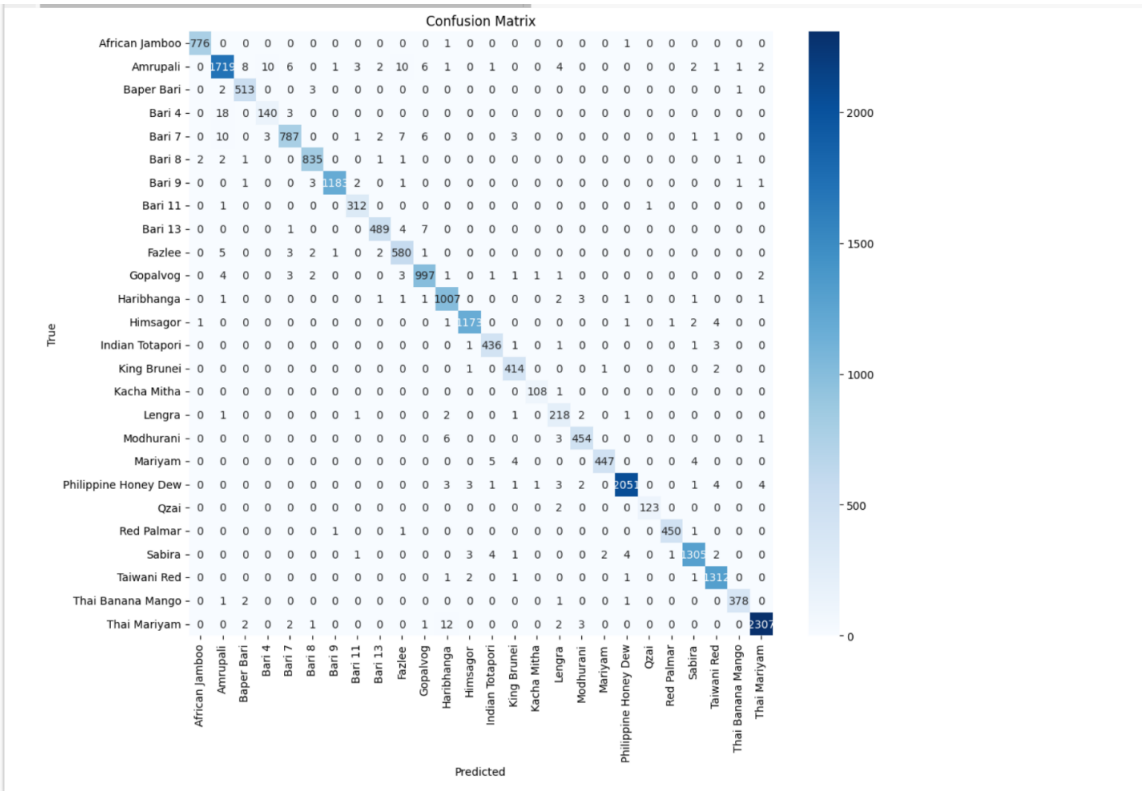



Figure 5.25: Confusion Matrix of Xception

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| African Jambo | 1.00 | 1.00 | 1.00 | 778 |
| Amrupali | 0.97 | 0.97 | 0.97 | 1777 |
| baper bari | 0.97 | 0.99 | 0.98 | 519 |
| Bari 4 | 0.92 | 0.87 | 0.89 | 161 |
| Bari 7 | 0.98 | 0.96 | 0.97 | 821 |
| Bari 8 | 0.99 | 0.99 | 0.99 | 843 |
| Bari 9 | 1.00 | 0.99 | 0.99 | 1192 |
| Bari 11 | 0.97 | 0.99 | 0.98 | 314 |
| Bari 13 | 0.98 | 0.98 | 0.98 | 501 |
| Fazlee | 0.95 | 0.98 | 0.97 | 594 |
| GopalVog | 0.98 | 0.98 | 0.98 | 1016 |
| Hari-Bhanga | 0.97 | 0.99 | 0.98 | 1019 |
| Himsagor | 0.99 | 0.99 | 0.99 | 1183 |
| Indian Totapori | 0.97 | 0.98 | 0.98 | 443 |
| King Brunei | 0.97 | 0.99 | 0.98 | 418 |
| Kacha Mitha | 0.98 | 0.99 | 0.99 | 109 |
| Modhurani | 0.92 | 0.96 | 0.94 | 226 |
| Moriyam | 0.98 | 0.98 | 0.98 | 464 |
| Lengra | 0.99 | 0.97 | 0.98 | 460 |
| Philippine Honey Dew | 1.00 | 0.99 | 0.99 | 2074 |
| Qzai | 0.99 | 0.98 | 0.99 | 125 |
| Red Palmar | 1.00 | 0.99 | 0.99 | 453 |
| Sabira | 0.99 | 0.99 | 0.99 | 1323 |
| Taiwani red | 0.99 | 1.00 | 0.99 | 1318 |
| Thai banana Mango Raw | 0.99 | 0.99 | 0.99 | 383 |
| Thai Morium | 1.00 | 0.99 | 0.99 | 2330 |
| accuracy | | | 0.98 | 20844 |
| macro avg | 0.98 | 0.98 | 0.98 | 20844 |
| weighted avg | 0.98 | 0.98 | 0.98 | 20844 |

Figure 5.26: Classification of Xception

The classification report for the Xception model is shown in Table 5.6, which includes important metrics covering precision, recall, F1-score, and support for the particular dataset. The report provides details on the model's performance using macro and weighted averages across the key measures, demonstrating a remarkable F1-score of 0.98.

Table 5.6: Classification Report of Xception

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.98 | 20844 |
| Macro Avg | 0.98 | 0.98 | 0.98 | 20844 |
| Weighted Avg | 0.98 | 0.98 | 0.98 | 20844 |

## 5.7 ResNet-50

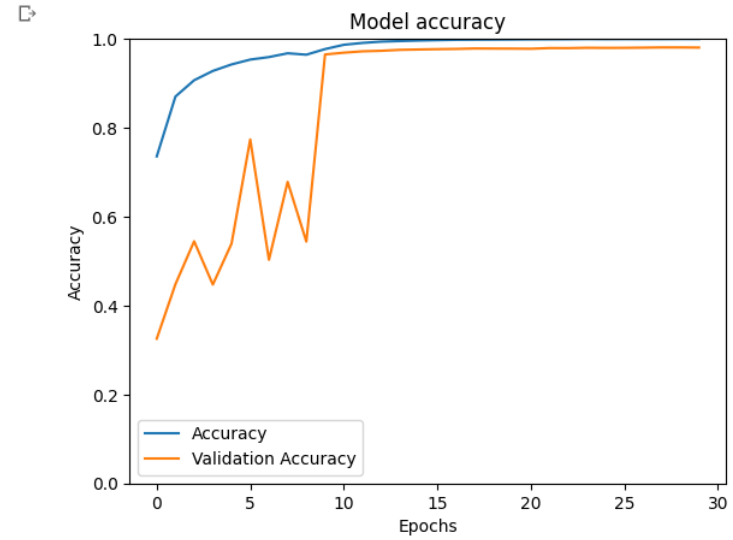



Figure 5.27: Model Accuracy graph of ResNet-50

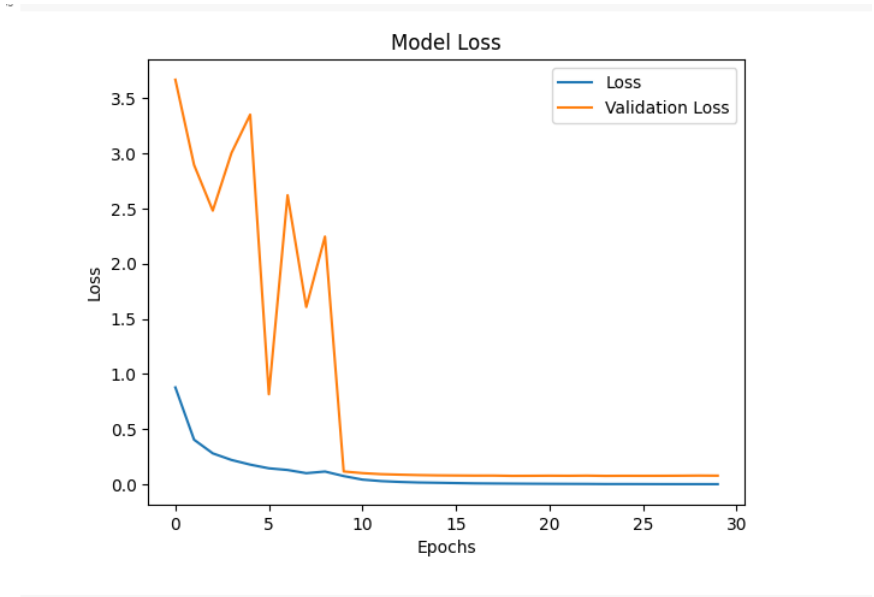



Figure 5.28: Model Loss graph of ResNet-50

In the above figure, it is shown that the accuracy curve increases as the number of epochs increases, which indicates that the model is learning and improving over time. The accuracy curve eventually converges to a value of around 98.10%, which is a very high accuracy. Other hand, a lower loss indicates that the model is more accurate. The loss curve decreases as the number of epochs increases, which indicates that the model is learning and improving over time. The loss curve eventually

converges to a value of around 0.2, which is a relatively low loss. If the loss curve starts to increase after a certain number of epochs, it is a sign that the model is overfitting Overall, the accuracy curve and the loss curve both show that the model is learning and improving over time. However, it is important to monitor the loss curve to make sure that it does not start to increase.

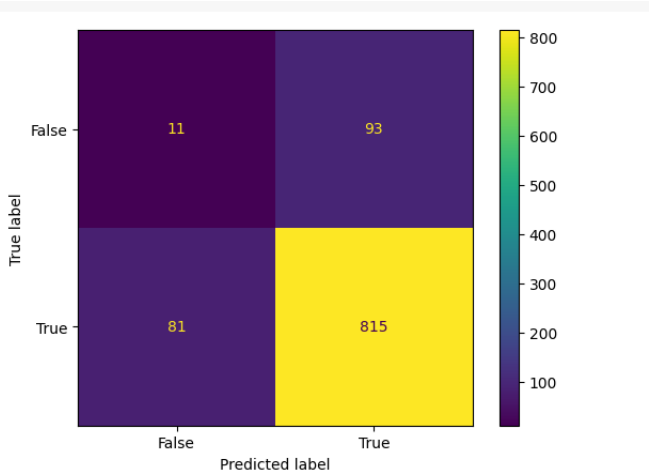



Figure 5.29: Predicted Table of ResNet-50

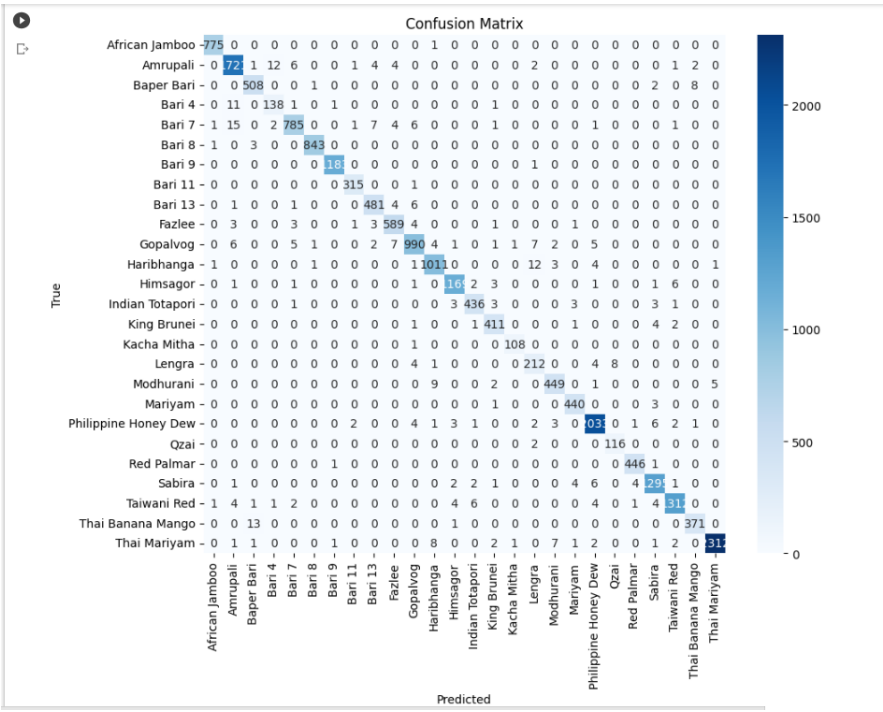



Figure 5.30: Confusion Matrix of ResNet-50

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| African Jambo | 0.99 | 1.00 | 1.00 | 776 |
| Amrupali | 0.98 | 0.98 | 0.98 | 1754 |
| baper bari | 0.96 | 0.98 | 0.97 | 519 |
| Bari 4 | 0.90 | 0.91 | 0.90 | 152 |
| Bari 7 | 0.98 | 0.95 | 0.96 | 824 |
| Bari 8 | 1.00 | 1.00 | 1.00 | 847 |
| Bari 9 | 1.00 | 1.00 | 1.00 | 1184 |
| Bari 11 | 0.98 | 1.00 | 0.99 | 316 |
| Bari 13 | 0.97 | 0.98 | 0.97 | 493 |
| Fazlee | 0.97 | 0.97 | 0.97 | 605 |
| GopalVog | 0.97 | 0.96 | 0.97 | 1032 |
| Hari-Bhanga | 0.98 | 0.98 | 0.98 | 1034 |
| Himsagor | 0.99 | 0.99 | 0.99 | 1185 |
| Indian Totapori | 0.97 | 0.97 | 0.97 | 450 |
| King Brunei | 0.96 | 0.98 | 0.97 | 420 |
| Kacha Mitha | 0.98 | 0.99 | 0.99 | 109 |
| Modhurani | 0.89 | 0.93 | 0.91 | 229 |
| Moriyam | 0.97 | 0.96 | 0.97 | 466 |
| Lengra | 0.98 | 0.99 | 0.98 | 444 |
| Philippine Honey Dew | 0.99 | 0.99 | 0.99 | 2059 |
| Qzai | 0.94 | 0.98 | 0.96 | 118 |
| Red Palmar | 0.99 | 1.00 | 0.99 | 448 |
| Sabira | 0.98 | 0.98 | 0.98 | 1316 |
| Taiwani red | 0.99 | 0.98 | 0.98 | 1340 |
| Thai banana Mango Raw | 0.97 | 0.96 | 0.97 | 385 |
| Thai Morium | 1.00 | 0.99 | 0.99 | 2339 |
| accuracy | | | 0.98 | 20844 |
| macro avg | 0.97 | 0.98 | 0.97 | 20844 |
| weighted avg | 0.98 | 0.98 | 0.98 | 20844 |

Figure 5.31: Classification of ResNet-50

Table 5.7 showcases the classification report for the ResNet-50 model, presenting essential metrics such as precision, recall, F1-score, and support for the given dataset. Furthermore, the report underscores a high F1-score of 0.98, providing a comprehensive view of the model's performance through macro and weighted averages across these metrics.

Table 5.7: Classification Report of ResNet-50

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.98 | 20844 |
| Macro Avg | 0.97 | 0.98 | 0.97 | 20844 |
| Weighted Avg | 0.98 | 0.98 | 0.98 | 20844 |

## 5.8 ViT

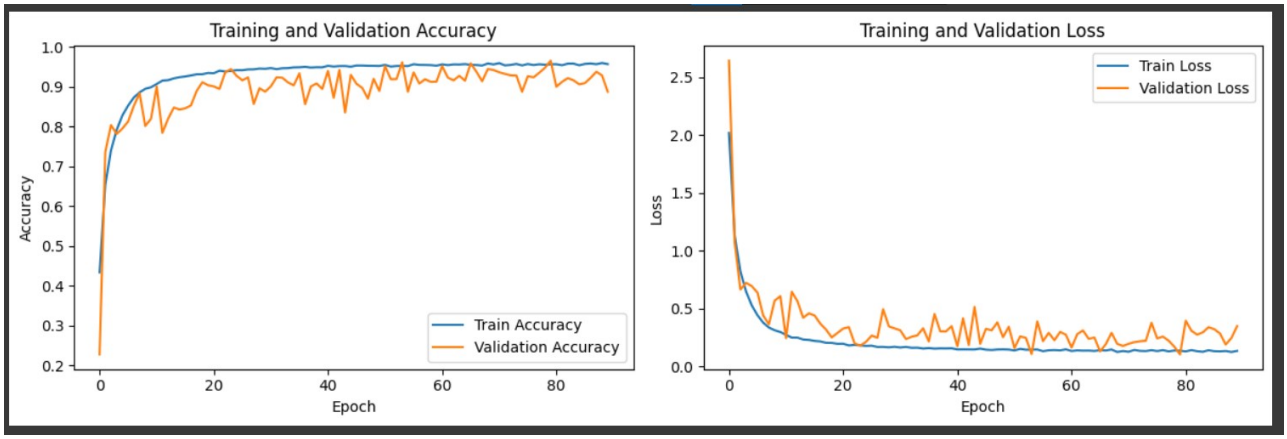



Figure 5.32: Model Accuracy and Loss graph of Vit

In this figure, the accuracy curve starts off low and then increases gradually. This indicates that the model is learning and improving over time. The accuracy curve eventually converges to a value of around 97%, which is a good accuracy. Moreover, the loss curve starts off high and then decreases gradually. This indicates that the model is learning and improving over time. The loss curve eventually converges to a value of around 0.45.

```
587/587 [==============================] - 50s 85ms/step - loss: 0.1340 - accuracy: 0.9577 - top-5-accuracy: 0.9984 - val_loss: 0.3241 - val_accuracy: 0.9096 - val_top-5-accuracy: 0.9978
Epoch 87/90
587/587 [==============================] - 48s 82ms/step - loss: 0.1325 - accuracy: 0.9587 - top-5-accuracy: 0.9990 - val_loss: 0.2873 - val_accuracy: 0.9228 - val_top-5-accuracy: 0.9906
Epoch 88/90
587/587 [==============================] - 52s 88ms/step - loss: 0.1358 - accuracy: 0.9570 - top-5-accuracy: 0.9987 - val_loss: 0.1919 - val_accuracy: 0.9381 - val_top-5-accuracy: 0.9978
Epoch 89/90
587/587 [==============================] - 51s 87ms/step - loss: 0.1272 - accuracy: 0.9600 - top-5-accuracy: 0.9986 - val_loss: 0.2485 - val_accuracy: 0.9288 - val_top-5-accuracy: 0.9964
Epoch 90/90
587/587 [==============================] - 49s 83ms/step - loss: 0.1367 - accuracy: 0.9572 - top-5-accuracy: 0.9987 - val_loss: 0.3510 - val_accuracy: 0.8877 - val_top-5-accuracy: 0.9928
217/217 [==============================] - 5s 24ms/step - loss: 0.0832 - accuracy: 0.9748 - top-5-accuracy: 0.9991
Test accuracy: 97.48%
Test top 5 accuracy: 99.91%
217/217 [==============================] - 7s 25ms/step
Weighted F1 score: 0.9749089598808541
Classification Report:

                     precision    recall  f1-score   support

       African Jambo       1.00      1.00      1.00       259
            Amrupali       0.99      0.96      0.97       588
             Bari 11       0.93      0.94      0.94       175
             Bari 13       0.84      0.94      0.89        51
              Bari 4       0.91      0.96      0.93       268
              Bari 7       1.00      0.98      0.99       282
              Bari 8       0.99      1.00      0.99       395
              Bari 9       1.00      0.98      0.99       106
              Fazlee       0.96      0.99      0.98       165
           GopalVog        0.97      0.97      0.97       202
         Hari-Bhanga       0.94      0.98      0.96       339
            Himsagor       0.93      0.97      0.95       345
     Indian Totapori       0.99      0.98      0.99       394
        Kacha Mitha        0.98      0.94      0.96       150
        King Brunei        0.98      0.96      0.97       142
            Lengra         1.00      1.00      1.00        36
          Modhurani        0.96      0.94      0.95        79
           Moriyam         0.98      0.94      0.96       154
 Philippine Honey Dew      0.91      0.99      0.95       150
              Qzai         0.98      0.99      0.99       687
         Red Palmar        0.97      0.95      0.96        41
            Sabira         1.00      1.00      1.00       150
         Taiwani red       0.99      0.97      0.98       439
         Thai Morium       0.99      1.00      0.99       442
Thai banana Mango Raw      0.94      0.93      0.94       127
         baper bari        0.99      0.97      0.98       772

           accuracy                            0.97      6938
          macro avg        0.97      0.97      0.97      6938
       weighted avg        0.98      0.97      0.97      6938
```

Figure 5.33: Classification of ViT

Here , the following table 5.8 displays the classification report for the Vision Transformer (ViT) model. The report provides key metrics such as precision, recall, F1-score, and support for the specific dataset. The F1-score is noted as 0.97, indicating a strong balance between precision and recall. The report also includes support values for each class, offering a detailed understanding of the model's performance. This comprehensive evaluation using macro and weighted averages across metrics provides a thorough assessment of the ViT model's effectiveness in classification.

Table 5.8: Classification Report of ViT

| | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| Accuracy | | | 0.97 | 6938 |
| Macro Avg | 0.97 | 0.97 | 0.97 | 6938 |
| Weighted Avg | 0.97 | 0.97 | 0.97 | 6938 |

## 5.9 Final analysis report among the Architectures

From the individual classification of all the models, it is noticeable that InceptionV3 performed the best with an accuracy of 98.9%. MobileNetV2, VGG16 , Xception, ResNet-50 and ViT models also performed well with an accuracy of 97.9%, 98.64%,98.42%,98.10% and 97% respectively. EfficientNetB3 performed relatively lower with an accuracy of 87.19%. Except for EfficientB3, All of our CNN models performed really well. ViT also performed really well and could be considered as an alternative model. The comparison between the models is visualized in the figure below.

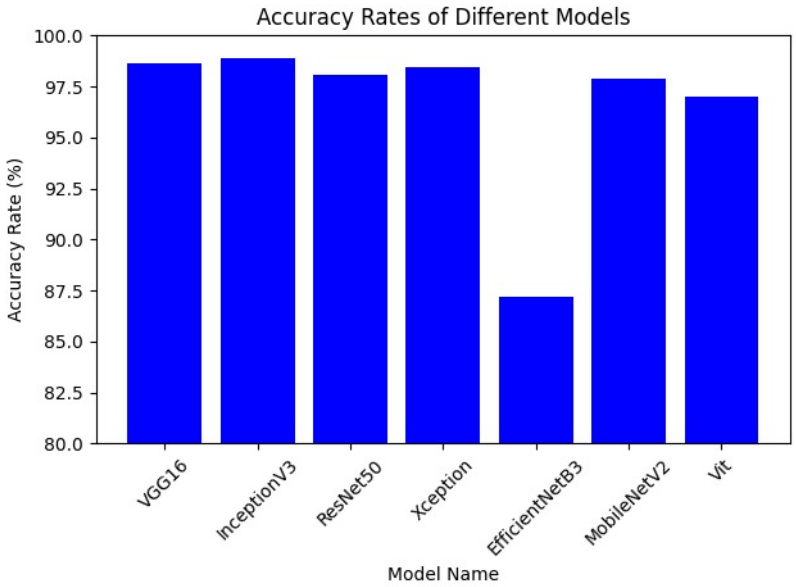



Figure 5.34: Accuracy Rate of Different Models

Table 5.9: Table of summary of the result analysis

| **Model** | **Accuracy Score** | **f-1 Score** | **Recall score** | **Precession Score** |
|---|---|---|---|---|
| **VGG16** | 98.64% | 98.64752% | 98.64709% | 98.65039% |
| **EfficientNetB3** | 87. 19% | 87.19354% | 87.19055% | 88.27344% |
| **MobileNetV2** | 97.9% | 97.90883% | 97.90827% | 97.91857% |
| **InceptionV3** | 98.89% | 98.89653% | 89.34465% | 89.90004% |
| **Xception** | 98.42% | 98.41670% | 98.41681% | 98.42363% |
| **ResNet-50** | 98.10% | 98.10640% | 98.10640% | 98.11277% |
| **ViT** | 97% | 97% | 97% | 98% |

The table 5.9 compares the accuracy and training time of top-performing algorithms for a binary classification dataset. Well-known models such as VGG16, EfficientNetB3, MobileNetV2, InceptionV3, Xception, ResNet-50, and ViT are evaluated using accuracy, f-1 score, recall score, and precision score, indicating their respective strengths in binary classification tasks.

# Chapter 6

# Conclusion and Future Work

## 6.1 Conclusion

In the end, this research has dealt with the significance of using deep learning models to automatically identify and classify mango varieties. By using well-known Convolutional Neural Networks (CNNs) and Vision Transformer (ViT) models, our suggested system has shown that it can distinguish between different types of mango with high accuracy and efficiency. Moreover, we have a huge collection of nearly 14,000 pictures of mango leaves from places as diverse as Rajshahi, Bogura, Feni, and Chapainawabganj. These images are like our teachers, helping us teach computers to recognize different types of mango leaves. As shown by the accuracy curves and loss curves, the results of our tests show that the models converged well and didn't overfit too much. This shows that the system can learn differentiating features from the mango pictures and apply them well to samples it hasn't seen before. Also, the confusion matrix has given us useful information about how well the model works. This has helped us find ways to improve the model and fix any misclassifications or confusion between different mango varieties. The detailed performance measures for each mango variety, such as precision, recall, F1-score, and support, were shown in the comprehensive classification report. With the help of these metrics, we were able to figure out how well the system did at correctly classifying the different kinds of mango, giving stakeholders reliable information for making decisions about quality control and inventory management.

In today's world, where we need to produce more food more efficiently, our research is extremely important. This shows that complex computer models like CNN and Vision Transformers can greatly help farmers in growing mangoes and other crops better. Our huge stock of photos of mango leaves, collected from different locations and different seasons, shows the real challenges farmers face today. Using these smart computer models and lots of imagery, we're giving agriculture a way to be more efficient and produce more food. In the future, we see that technology and agriculture will become close friends in solving global food problems. Our research guides us to use powerful computational models, such as VGG16, InceptionV3, ResNet50, Xception, EfficiencyNetB3, MobileNetV2, and Vision Transformer (ViT) to find smart solutions for better agriculture.

The developed system will have a lot of real-world effects on the agriculture business. It can help farmers, distributors, and sellers make sure that mango varieties are correctly identified and put into groups. This improves inventory management and

quality control. Also, putting the system into automatic sorting processes could make it easier to sort and package mangoes, making them more efficient and reducing the amount of work that needs to be done by hand. This thesis has mostly been about identifying and classifying mango varieties, but the suggested method can also be used for other fruits and vegetables, giving automated classification a wider range of uses. By using deep learning models and transfer learning techniques, our system shows how current technologies have the potential to change the way farming is done and make it more productive and profitable overall.
Finally, our work serves as both a contribution to the field of mango leaf variety identification and an important call for the modernization of agriculture. This study worked to improve automated methods for classifying fruits, especially when it comes to identifying and classifying mango varieties. When deep learning models, accuracy curves, loss curves, the confusion matrix, and the classification report are all put together, they give a full picture of how the system works and what it means in real life. By automating the identification and classification process, we can make it possible to improve quality control, inventory management, and market analysis in the agriculture industry.

## 6.2 Future Work

While this research marks an important milestone in the field of automated fruit sorting, it also opens the door to a series of exciting opportunities for future exploration. First, expanding our dataset to include more mango samples, accounting for regional variations, and incorporating multi-seasonal data, is paramount to enhance adaptability and reliability system reliability. Second, ensemble model mining, which harnesses the collective power of diverse architectures, promises to push classification accuracy to new heights. Developing real-time deployments for agricultural environments, such as packinghouses and distribution centers, represents a real-world evolution of our work, turning it into tangible benefits for agriculture. people and supply chains. In addition to mangoes, extending our method to classify other fruits and vegetables promises to meet broader agricultural needs. Integrating human expertise into systems to facilitate continuous improvement and adaptability as well as assess sustainability is critical to a comprehensive approach to automation agricultural chemistry. Additionally, practical implementation strategies and user-friendly interfaces will play a key role in ensuring our technology is seamlessly integrated into real-world agricultural operations. In short, this research not only advances automated fruit sorting but also signifies a broader transformation in agricultural practices, towards a better, more efficient future. As technology continues to evolve, the fusion of deep learning models and automation in agriculture promises increased efficiency, reduced waste, and heightened food security. The future avenues outlined herein represent the next logical steps in realizing this potential and furthering the impact of automated fruit classification within the agricultural industry.

# Bibliography


[1] A. Rocha, D. C. Hauagge, J. Wainer, and S. Goldenstein, "Automatic fruit and vegetable classification from images," *Computers and Electronics in Agriculture*, vol. 70, no. 1, pp. 96–104, 2010.

[2] J. Chaki and R. Parekh, "Plant leaf recognition using shape based features and neural network classifiers," *International Journal of Advanced Computer Science and Applications*, vol. 2, no. 10, 2011.

[3] J. F. S. Gomes and F. R. Leta, "Applications of computer vision techniques in the agriculture and food industry: A review," *European Food Research and Technology*, vol. 235, pp. 989–1000, 2012.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[5] C. S. Nandi, B. Tudu, and C. Koley, "Computer vision based mango fruit grading system," in *International Conference on Innovative Engineering Technologies (ICIET 2014) Dec*, 2014, pp. 28–29.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[7] A. Aakif and M. F. Khan, "Automatic classification of plants based on their leaves," *Biosystems Engineering*, vol. 139, pp. 66–75, 2015.

[8] I. Maqbool, S. Qadri, D. M. Khan, and M. Fahad, "Identification of mango leaves by using artificial intelligence," *International journal of natural and engineering sciences*, vol. 9, no. 3, pp. 45–53, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[10] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in plant science*, vol. 7, p. 1419, 2016.

[11] E. Prasetyo, "Detection of mango tree varieties based on image processing," *Indonesian journal of science and technology*, vol. 1, no. 2, pp. 203–215, 2016.

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[13] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[14] A. G. Howard, M. Zhu, B. Chen, *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] X. Ren, B. Kang, and Z. Zhang, "Understanding tumor ecosystems by single-cell sequencing: Promises and limitations," *Genome biology*, vol. 19, no. 1, pp. 1–14, 2018.

[16] N. O. Delgado, E. R. Arboleda, J. L. Dioses Jr, and R. M. Dellosa, "Identification of mango leaves using artificial intelligence," *Int. J. Sci. Technol. Res*, vol. 8, no. 12, pp. 2864–2868, 2019.

[17] W. M. Tabada and J. G. Beltran, "Mango variety recognizer using image processing and artificial neural network," 2019.

[18] I. Zarrin and s. Islam, "Leaf based trees identification using convolutional neural network," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 1–4. DOI: 10.1109/I2CT45611.2019.9033914.

[19] A. D. Supekar and M. Wakode, "Multi-parameter based mango grading using image processing and machine learning techniques," *INFOCOMP Journal of Computer Science*, vol. 19, no. 2, pp. 175–187, 2020.

[20] P. Wang, "Research and design of smart home speech recognition system based on deep learning," in *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2020, pp. 218–221. DOI: 10.1109/CVIDL51233.2020.00-98.

[21] P. Wei, S. Xia, R. Chen, J. Qian, C. Li, and X. Jiang, "A deep-reinforcement-learning-based recommender system for occupant-driven energy optimization in commercial buildings," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6402–6413, 2020. DOI: 10.1109/JIOT.2020.2974848.

[22] A. M. U. Zaman, "72 varieties of mangoes available in country," Jun. 2020.

[23] L. Alzubaidi, J. Zhang, A. J. Humaidi, *et al.*, "Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions," *Journal of big Data*, vol. 8, pp. 1–74, 2021.

[24] E. Bayhan, Z. Ozkan, M. Namdar, and A. Basgumus, "Deep learning based object detection and recognition of unmanned aerial vehicles," in *2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, IEEE, 2021, pp. 1–5.

[25] T. Hugo, M. Cord, D. Matthijs, M. Francisco, S. Alexandre, and J. Herve, "Training data-efficient image transformers & distillation through attention," in *ICML*, 2021.

[26] B. J. Hutasoit, H. Sofyan, and F. R. Kodong, "Classification of mango plants based on leaf shape using glcm and k-nearest neighbor methods," *Computing and Information Processing Letters*, vol. 1, no. 1, pp. 1–7, 2021.

[27] N. Nafi'iyah and J. Maknun, "Cnn architecture for classifying types of mango based on leaf images," *Telematika*, vol. 14, no. 2, pp. 112–121, 2021.

[28] A. Sarkar, "Understanding efficientnet-the most powerful cnn architecture," *Toronto: Medium. Available online at: https://medium. com/mlearning-ai/understanding-efficientnet-the-most-powerful-cnn-architecture-eaeb40386fad (accessed August 13, 2022)*, 2021.

[29] P. S. Thakur, P. Khanna, T. Sheorey, and A. Ojha, "Vision transformer for plant disease detection: Plantvit," in *International Conference on Computer Vision and Image Processing*, Springer, 2021, pp. 501–511.

[30] T. Aslam, S. Qadri, S. F. Qadri, *et al.*, "Machine learning approach for classification of mangifera indica leaves using digital image analysis," *International Journal of Food Properties*, vol. 25, no. 1, pp. 1987–1999, 2022.

[31] R. A. Boukabouya, A. Moussaoui, and M. Berrimi, "Vision transformer based models for plant disease detection and diagnosis," in *2022 5th International Symposium on Informatics and its Applications (ISIA)*, IEEE, 2022, pp. 1–6.

[32] N. Mahmud, "Mango wonder: 200 varieties in a single tree in chapainawabganj," 2022.

[33] A. Alotaibi, T. Alafif, F. Alkhilaiwi, *et al.*, "Vit-deit: An ensemble model for breast cancer histopathological images classification," in *2023 1st International Conference on Advanced Innovations in Smart Cities (ICAISC)*, 2023, pp. 1–6. DOI: 10.1109/ICAISC56366.2023.10085467.