Papaya Fruit Disease Classification using Deep Learning Techniques

A report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Department of Computer Science and Engineering

by

ABHISHEK TIRKEY [CS20B1002]
UJJWAL KUMAR [CS20B1060]
K VIVEK KUMAR [CS20B1065]

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY
KARAIKAL – 609609
MAY 2024

BONAFIDE CERTIFICATE

This is to certify that the project work phase I entitled **"Papaya Fruit Disease Classification using Deep Learning Techniques"** is a bonafide record of the work done by

ABHISHEK TIRKEY [CS20B1002]
UJJWAL KUMAR [CS20B1060]
K VIVEK KUMAR [CS20B1065]

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** of the **NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY** during the year 2023 - 2024.

DR. P KUMARAN
ASSISTANT PROFESSOR
Project Guide

Dr. Ansuman Mahapatra
Head of the Department
Computer Science & Engineering

Project viva-voice held on: 02 May 2024

Internal Examiner

External Examiner

ABSTRACT

The project aims to classify papaya fruit diseases using a deep learning model. This project aims to help in the field of Agriculture. In recent years, deep learning has emerged as a promising tool for image classification. This study investigates the potential of various deep learning models for papaya fruit disease classification. We evaluated eight pre-trained models: AlexNet, DenseNet, EfficientNet, InceptionNet, MobileNet, NASNet, ResNet, and VGGNet. We employed a fine-tuning approach specifically with MobileNet due to its promising initial performance on our dataset of papaya fruit images labeled with different disease categories and healthy fruit. Moreover, we proposed our CNN model to get robust model with high accuracy. The project's outcome will be an optimized deep learning model that can accurately classify papaya fruit diseases or identify healthy fruits and this would be helpful in the field of agriculture.

ACKNOWLEDGEMENT

We would like to show our kind regards towards our respected Director, **Dr. Usha Natesan** for permitting us to undertake this project work.

We would like to thank our project guide, **Dr. P. Kumaran** for his constant motivation and guidance during the project. We want to genuinely convey our thanks to project coordinator **Dr. Chandrasekar. R** and **Dr. Ansuman Mahapatra,** Head of the Department and all the faculties of our Computer Science and Engineering department for their motivation in various reviews throughout the course of the project phase-II.

We are at the dearth of words to express gratitude to our wonderful parents for their unconditional support both financially and emotionally. I thank our parents for inculcating the dedication and discipline to do whatever we undertake well.

We have been fortunate to have friends who cherish us despite our eccentricities. By their remarks, comments or compliments and unavoidable questions, we were able to make our project reviews better each time. Thank you all for making it possible for us to reach the final stage of our endeavor.

We would also like to thank all our sources, mentioned in the references, and our friends who helped us by providing mental and logistical support. Last but not the least we would like to thank our parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Papaya, a tropical fruit cherished for its delectable taste and nutritional value, faces a substantial threat from various diseases that can significantly diminish crop yields [1]. Conventional methods for disease detection often prove inefficient and imprecise. Papaya is a versatile fruit, consumed both ripe and unripe. Ripe papaya, with its soft, orange flesh, is enjoyed fresh or incorporated into various dishes, such as salads, smoothies, and desserts. Unripe papaya, with its firm, green flesh, is often used in savory preparations, including curries, pickles, and chutneys. Papaya's nutritional profile makes it a valuable addition to the Indian diet. It is a rich source of vitamin C, essential for immune function and collagen production, as well as vitamin A, crucial for vision and cell growth. Papaya also contains fiber, promoting digestive health, and potassium, regulating blood pressure [2].

Fig.1 Healthy Papaya Fruit

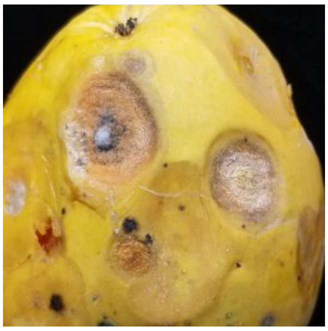

Fig.2 Diseased Papaya Fruit

Papaya fruit diseases, including Anthracnose, Phytophthora Blight, Brown spot, Black spot, threaten fruit quality and yield. Management involves disease-resistant varieties, sanitation, and fungicides for healthy papaya crops.

Anthracnose is a disease that affects papaya fruit, causing small, dried, pink spots on the surface of ripening fruit [3]. These spots can grow to 5 cm in diameter, become rounded and sunken, and turn brown to black in color. The lesions can be either water-soaked or dried and hard.

Phytophthora blight is a fungus-like disease that causes fruit rot, stem rot, root rot, and damping-off in papaya plants [4]. It's caused by the water mold Phytophthora palmivora, which is related to algae and not a fungus. The disease is most severe in wet, windy weather, and conditions that favor it include: Excessive soil moisture, Ambient temperatures of 20–30°C, and Fruit wetness periods of 72 hours or more.

Fig.3 Anthracnose

Fig.4 Phytophthora Blight

Fig.5 Brown Spot

Fig.6 Black Spot

Brown spot is a serious foliar disease that affects papaya leaves and fruit. Symptoms of brown spot include light brown circular spots on fruit [5]. Brown spot disease on papaya fruit is caused by a fungus called Corynespora cassiicola. This disease can cause significant losses in papaya production, especially in hot and humid conditions.

Black spot is a fungal disease that can affect papaya leaves and fruits [6]. The disease is caused by the fungus Asperisporium caricae, and the scientific name for the disease is Cercospora papayae. This disease is most severe during rainy periods.

In response to this challenge, our project is dedicated to harnessing the potential of deep learning to revolutionize papaya farming. We aim to develop a robust system employing advanced algorithms and data-driven techniques to accurately classify diseases that us been affected in papaya fruits as either healthy or diseased. This innovative solution has the potential to provide farmers with timely insights, enabling them to safeguard their crops and ensure more abundant yields.

Throughout this project, we will delve into the methodologies, tools, and outcomes of implementing deep learning for papaya fruit disease classification, thereby contributing to the progress of sustainable agriculture.

1.2 Definition

Papaya fruit disease classification using deep learning is a process that utilizes deep learning algorithms to analyze and classify images of papaya fruits to determine whether which diseases is been affected in the papaya fruit or the fruit is healthy. This approach aims to improve the accuracy and efficiency of disease classification, reduce crop losses due to disease, enhance yields and quality of papaya fruits, and minimize the reliance on chemical treatments. Papaya fruit disease classification refers to the identification and analysis of ailments affecting papaya plants and their fruit. Utilizing various technologies such as image recognition, molecular diagnostics, and data analysis, this process aims to swiftly and accurately diagnose diseases that can negatively impact papaya yield and quality. Early detection enables prompt intervention, including targeted treatments or crop management strategies, to mitigate the spread of diseases and safeguard papaya crops. This proactive approach contributes to sustainable agriculture by minimizing economic losses and ensuring the production of healthy papayas for consumption.

1.3 Objective

The objective of this major project is to develop and evaluate deep learning models for the disease’s classification in the papaya fruit. The project aims to improve the accuracy and efficiency of disease classification, reduce crop losses due to disease, improved yields and quality of papaya fruits and to reduce use of pesticides and other chemicals.

Overall, the objective of papaya fruit disease classification using deep learning is to develop tools that can help farmers to improve the sustainability and profitability of papaya production.

1.4 Motivation

Economic and agricultural impact:

Reduced crop loss: Papaya is a valuable crop grown in many tropical and subtropical regions. Diseases can significantly reduce papaya yields, impacting farmers' income and food security. Early and accurate disease detection can help farmers take timely action to control outbreaks and minimize losses.

Improved fruit quality: Deep learning models can be used to identify diseases at an early stage, before they significantly affect the fruit's quality. This allows farmers to harvest fruits at the optimal time and reduces post-harvest losses.

Advancements in technology:

Deep learning's effectiveness: Deep learning techniques have proven highly successful in image classification tasks. Their ability to learn complex patterns from image data makes them ideal for identifying diseases in fruits like papaya based on visual symptoms.

Potential for automation: A deep learning model for papaya disease classification could be integrated into smartphone apps or automated field monitoring systems. This would allow farmers to easily diagnose diseases in their fields without needing extensive expertise

Social and environmental benefits:

Reduced reliance on pesticides: Early disease detection can help farmers implement targeted control measures, reducing their reliance on broad-spectrum pesticides. This can benefit human health and the environment.

Sustainable agriculture: Deep learning-based disease classification can contribute to more sustainable agricultural practices by promoting early intervention and reducing reliance on chemical control methods.

Overall, a project on papaya fruit disease classification using deep learning techniques has the potential to address a significant challenge in papaya production, improve farmer livelihoods, and contribute to advancements in sustainable agriculture.

1.5 Applications

The application of papaya fruit disease classification using deep learning algorithms has the potential to revolutionize the papaya industry by reducing crop losses, enhancing fruit quality, and ensuring food safety.

- **Early disease detection**: The system can accurately detect papaya diseases at an early stage, even before visible symptoms appear. This allows farmers to take timely action to prevent the spread of disease and minimize crop losses.
- **Reduced reliance on fungicides**: By providing early detection of diseases, the system can help farmers reduce their reliance on fungicides, which can be harmful to the environment and human health.
- **Improved yield and quality**: By preventing diseases and optimizing crop management, the system can help farmers increase the yield and quality of their papaya crops.
- **Enhanced farmer decision-making:** The system can provide farmers with valuable insights into the health of their crops, enabling them to make better decisions about irrigation, fertilization, and pest control.
- **Sustainability**: By promoting sustainable practices and reducing the use of harmful chemicals, the system can contribute to the sustainability of papaya farming.
- **Reduced Costs**: Early detection and targeted treatments can lead to cost savings for farmers. By using deep learning techniques, the project aims to reduce the need for manual inspection and intervention, saving both time and resources.
- **Research and Development**: The project can contribute to the advancement of research in agricultural technology and deep learning applications. It can serve as a basis for further studies on other crops and diseases, leading to innovations in agricultural practices.

CHAPTER 2

SYSTEM REQUIREMENT SPECIFICATIONS

2.1 Hardware requirement

Processing Unit (CPU/GPU): A powerful CPU or GPU is essential for handling the computational demands of machine learning algorithms, especially for processing large image datasets. A high-end CPU or a mid-range GPU is recommended for training and running papaya fruit disease detection models.

Memory (RAM): Sufficient RAM is crucial for loading large image datasets and training complex models. A minimum of 8GB of RAM is recommended, but 16GB or more is preferred for more demanding tasks.

Storage: Adequate storage space is necessary for storing image datasets, training models, and intermediate results. A minimum of 1GB of storage is recommended, but more may be needed depending on the dataset size and model complexity.

2.2 Software requirement

Programming Language: Python is the most widely used programming language for machine learning and is recommended for developing papaya fruit disease detection algorithms.

Deep Learning Frameworks: AlexNet, DenseNet, EfficientNet, InceptionNet, MobileNet, NASNet, ResNet, VGGNet are popular deep learning architectures used for image recognition, classification, and other computer vision tasks. These frameworks offer pre-built models or serve as a foundation for building your own custom deep neural networks.

Image Processing Libraries: OpenCV, Matplotlib, Seaborn and Pillow are commonly used image processing libraries that provide functions for image manipulation, preprocessing, and feature extraction.

Development Environment: An integrated development environment (IDE) such as PyCharm or Jupyter Notebook can provide a convenient platform for writing, debugging, and executing machine learning code.

Model Deployment Tools: If deploying the model to a production environment, tools like Streamlit, can be used to create lightweight and efficient inference pipelines.

CHAPTER 3

LITERATURE SURVEY

3.1 Representing Image

In this comprehensive dataset, a diverse collection of images depicting both healthy and diseased papaya fruits have been meticulously curated. In my dataset, I categorize images into six distinct classes: Healthy Papaya Fruit, Anthracnose, Black Spot, Brown Spot, Phytophthora Blight, and Others. The first five classes represent specific diseases or conditions affecting papaya fruit, including Anthracnose, Black Spot, Brown Spot, and Phytophthora Blight. The sixth class, Others, encompasses any images that do not fall into the aforementioned categories. This classification scheme allows for the comprehensive representation of various states and conditions of papaya fruit, facilitating accurate analysis and diagnosis within the dataset.

3.2 Previous Research

Several studies have explored the use of DL for papaya fruit disease classification. These studies have employed various DL techniques, including convolutional neural networks (CNNs) various pretrained model to classify the papaya fruit using deep learning.

One study, titled "Yolo-Papaya: A Papaya Fruit Disease Detector and Classifier Using CNNs and Convolutional Block Attention Modules," [7] proposed a novel Yolo-Papaya model that achieved a mean average precision (mAP) of 86.2% for disease detection. This high accuracy demonstrates the potential of CNNs for papaya fruit disease detection.

Another study, titled "Machine vision-based papaya disease recognition," [8] proposed a system that utilizes k-means clustering to segment out diseased regions from papaya fruit images and then extracts texture features from these regions for classification using an SVM classifier. This system achieved an accuracy of 92.3% for disease classification.

These studies demonstrate the effectiveness of ML-based approaches for papaya fruit disease detection. ML algorithms can accurately classify papaya fruit images into healthy and diseased categories, which can aid in early disease detection and prevention measures. As ML techniques continue to evolve, we can expect further advancements in papaya fruit disease detection accuracy and efficiency. Automated Detection of Papaya Ringspot Virus in Papaya Plants Using Image Processing Techniques: This study explores the use of image processing techniques to automatically detect symptoms of papaya ringspot virus in papaya plants. Molecular Techniques for the Detection and Characterization of Papaya Diseases: This study investigates molecular diagnostic tools for accurate identification and characterization of various papaya diseases.

Machine Learning Approaches for Early Detection of Papaya Bacterial Blight: [9] This study examines the application of machine learning algorithms for early detection of bacterial blight in papaya plants. Remote Sensing and GIS-Based Monitoring of Papaya Black Spot Disease: This study utilizes remote sensing and geographic information systems (GIS) to monitor and manage papaya black spot disease. Sensor Technologies for Real-Time Monitoring of Papaya Fungal Diseases: This study investigates the use of sensor technologies for continuous, real-time monitoring of fungal diseases affecting papaya. Biological Control Strategies for Papaya Anthracnose: This study explores biological control methods for managing anthracnose in papaya, providing an eco-friendly alternative to chemical treatments. Genomic Approaches to Understanding Papaya Resistance to Viral Infections: This study examines the genetic factors contributing to papaya resistance against viral infections, with potential implications for breeding disease-resistant varieties.

Hossen [10] proposed a deep neural network (DNN) model for classifying papaya fruits as “diseased” or “healthy”. The study employed a dataset of 234 images, with 184 images used for training, 28 for validation, and 22 for testing. The proposed network consisted of a basic convolutional neural network (CNN) with three convolutional layers followed by max pooling and two dense layers with a sigmoid function in the classification function. Although examples of both healthy and diseased papaya fruits and leaves were presented, the number of images used to form the training and test sets of leaves and fruits were not specified. The authors reported an average accuracy of 91% for the classification task, which is remarkable given the limited number of training examples in a CNN network. The source code and images utilized in the study were not made publicly available

In a similar study, Hossen [11] also compared several algorithms for the classification of five diseases (“anthracnose”,” black spot”, “Phytophthora”, “powdery mildew”, and “ring spot”) in papaya fruits. The study compared the performance of random forest, kmeans clustering, support vector machine (SVM), and convolutional neural network (CNN) classifiers. The dataset used for the experiments consisted of 214 images, with 128 images utilized for training and 86 for testing. No validation set was used in the study. The CNN approach achieved the highest accuracy with 98%. The absence of a validation set, and the unavailability of source code and dataset, make reproducing the reported results difficult.

Integrated Pest Management for Sustainable Control of Papaya Mealybug [12]: This study discusses integrated pest management strategies, including biological control and cultural practices, for sustainable control of the papaya mealybug. Smart Farming Technologies for Early Detection of Papaya Diseases [13]: This study explores the integration of smart farming technologies, such as IoT devices and data analytics, for early detection and management of papaya diseases. Epidemiological Modeling of Papaya Disease Spread [14]: This study focuses on mathematical modeling to understand and predict the spread of papaya diseases, aiding in the development of effective control strategies.

3.3 Challenges and Limitations

Data availability: There is a lack of large, publicly available datasets of papaya fruit images with labeled disease annotations. This makes it difficult to train and evaluate machine learning models for disease detection. Fruit variability: Papaya fruits can vary greatly in appearance due to factors such as cultivar, ripeness, and environmental conditions. This can make it difficult to develop detection algorithms that are generalizable to different papaya varieties and growing conditions.

Early detection: Some papaya diseases are difficult to detect early on, when they are most treatable. This can lead to yield losses and economic hardship for papaya growers. Disease complexity: Papaya fruits can be infected with multiple diseases at the same time. This can make it difficult to accurately diagnose diseases using machine learning models.

Real-time detection: There is a need for real-time detection of papaya fruit diseases to facilitate timely intervention and prevent the spread of disease. However, current detection methods are often slow and computationally expensive. Despite these challenges, there is significant progress being made in the development of papaya fruit disease detection technologies. Machine learning, image processing, and sensor technologies are all being used to develop new methods for detecting and classifying papaya fruit diseases. As these technologies continue to develop, we can expect to see more accurate, efficient, and real-time methods for papaya fruit disease detection.

3.4 Future Directions

In the future, this project holds promising avenues for advancement. Expanding the dataset to include a more extensive array of papaya diseases and diverse environmental conditions will enhance the model's ability to generalize. Integrating additional advanced algorithms or delving into deep learning models could offer insights into alternative approaches for disease detection. Transitioning towards real-time detection capabilities and developing a user-friendly mobile application for on-site analysis using smartphones would empower farmers with immediate feedback. Continuous monitoring, regular updates with new data, and collaboration with agricultural experts will be pivotal for maintaining the model's relevance in dynamic agricultural landscapes. Additionally, exploring transfer learning and customizing the model for specific regions can further boost its accuracy and applicability. The iterative improvement process should be guided by user feedback, particularly from farmers, ensuring practicality and relevance. Ultimately, these future directions aim to transform the project into a versatile and robust tool, contributing significantly to effective papaya disease detection and crop management in diverse agricultural settings.

CHAPTER 4

IMPLEMENTATION & RESULTS

4.1 Model Architecture

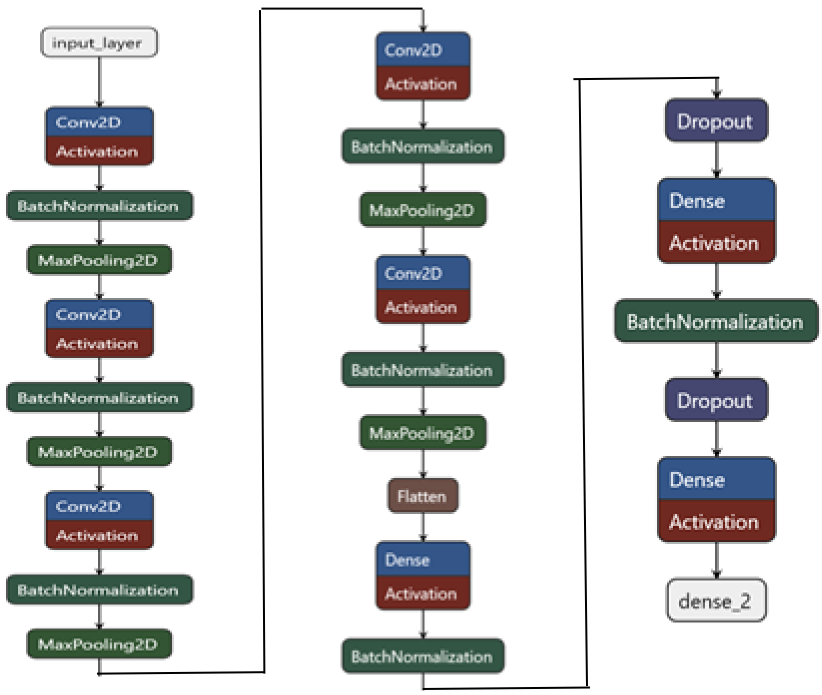

Fig.7 Model Architecture

4.2 Data Curation and Data Pre-Processing

A dataset of papaya fruit images is being collected to distinguish between healthy and diseased fruits. Initially, images containing multiple fruits were removed to ensure each image represents a single papaya. The remaining images were then equally distributed across six classes: healthy, anthracnose, phytophthora blight, brown spot, black spot, and others. Each class contains 650 images for training a deep learning model to identify these diseases, and an additional 100 images for testing the model's accuracy.

4.3 Pre-Trained Models

To achieve high accuracy efficiently, we evaluated eight pre-trained models and fine-tuned the one with the best training performance.

4.3.1AlexNet:

AlexNet is a convolutional neural network (CNN) architecture that was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton in 2012 [14] . It is considered to be a breakthrough in the field of computer vision, as it was the first CNN to win the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. AlexNet is a deep convolutional neural network that consists of eight layers: five convolutional layers, two fully connected layers, and one output layer. The convolutional layers are responsible for extracting features from the input images, while the fully connected layers are responsible for classifying the images.

The first convolutional layer of AlexNet consists of 96 filters of size 11x11 with a stride of 4. This layer is followed by a max pooling layer with a pool size of 3x3 and a stride of 2. The second convolutional layer consists of 256 filters of size 5x5 with a stride of 1. This layer is also followed by a max pooling layer with a pool size of 3x3 and a stride of 2. The third, fourth, and fifth convolutional layers consist of 384, 384, and 256 filters of size 3x3 with a stride of 1, respectively. These layers are all followed by a max pooling layer with a pool size of 3x3 and a stride of 2.

The fully connected layers of AlexNet consist of 4096 and 4096 neurons, respectively. The output layer consists of 1000 neurons, which corresponds to the 1000 classes in the ImageNet dataset. AlexNet was trained on the ImageNet dataset, which consists of over 14 million images labeled with 1000 different classes. AlexNet achieved a top-5 test accuracy of 15.3% on the ImageNet dataset, which was a significant improvement over the previous state-of-the-art of 26.2%. AlexNet has a number of benefits that make it a powerful CNN architecture. First, AlexNet is a deep CNN, which means that it has a large number of layers. This allows AlexNet to learn complex features from the input images. Second, AlexNet uses a variety of techniques to improve its performance, such as dropout regularization and data augmentation. Third, AlexNet was trained on a large dataset, which allows it to learn a wide range of features.

AlexNet also has a number of limitations. First, AlexNet is a computationally expensive model to train and deploy. Second, AlexNet is not as robust to noise and occlusions as some other CNN architectures. Third, AlexNet is not as scalable as some other CNN architectures. Despite its limitations, AlexNet is an important CNN architecture that has had a significant impact on the field of computer vision. AlexNet has demonstrated the power of deep CNNs for image classification and has inspired the development of many other successful CNN architectures.

4.3.2 Dense Net:

DenseNet, or Dense Convolutional Network [15], is a type of convolutional neural network that uses dense connectivity between layers. This means that each layer is connected to all previous layers, which allows for better feature reuse and gradient flow. DenseNets have been shown to achieve state-of-the-art results on a variety of image classification tasks.

One of the key benefits of DenseNets is that they are very efficient in terms of parameters. This is because each layer only needs to learn a small number of new features, since it has access to all of the features from previous layers. This makes DenseNets well-suited for tasks where the dataset is small or where the model needs to be deployed on a device with limited resources.

Another benefit of DenseNets is that they are very easy to train. This is because the dense connectivity between layers allows for gradients to flow more easily through the network. This makes DenseNets less likely to suffer from the vanishing gradient problem, which can make it difficult to train deep neural networks.

DenseNets also have a number of limitations. One limitation is that they can be computationally expensive to train, especially for large datasets. This is because each layer needs to be connected to all previous layers, which can lead to a large number of parameters.

Another limitation of DenseNets is that they can be prone to overfitting. This is because the dense connectivity between layers can allow the network to learn too much from the training data, which can lead to poor performance on new data.

Overall, DenseNets are a powerful type of convolutional neural network that offer a number of benefits, including parameter efficiency, ease of training, and improved performance. However, DenseNets can also be computationally expensive to train and prone to overfitting.

4.3.3 EfficientNet:

EfficientNet is a convolutional neural network [16] architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. It is a product of Compound Scaling and Neural Architecture Search (NAS). EfficientNet aims to improve performance while increasing computational efficiency by reducing the number of parameters and FLOPs (Floating point Operations Per Second).

EfficientNet uses Mobile Inverted Bottleneck (MBConv) layers, which are a combination of depth-wise separable convolutions and inverted residual blocks. MBConv layers are more efficient than traditional convolutional layers because they reduce the number of parameters and FLOPs required to perform the same operation.

EfficientNet has two parts:

1. Create an efficient baseline architecture using NAS
2. Use the Compound Scaling Method while scaling up to enhance performance

The first part of EfficientNet uses NAS to create an efficient baseline architecture. NAS is a technique for automatically searching for neural network architectures that are both accurate and efficient. The second part of EfficientNet uses the Compound Scaling Method to scale up the baseline architecture while maintaining its efficiency.

The Compound Scaling Method is a novel technique for scaling up neural networks. It works by uniformly scaling all dimensions of the network (depth, width, and resolution) by a compound coefficient. This approach allows EfficientNet to achieve state-of-the-art accuracy on a variety of image classification datasets while using fewer parameters and FLOPs than other convolutional neural networks.

EfficientNet has several benefits over other convolutional neural networks. First, it is more efficient, meaning that it requires fewer parameters and FLOPs to achieve the same accuracy. Second, it is more accurate, meaning that it achieves higher accuracy on a variety of image classification datasets. Third, it is more scalable, meaning that it can be scaled up to larger sizes without sacrificing accuracy or efficiency. However, EfficientNet also has some limitations. First, it is more complex than other convolutional neural networks, making it more difficult to train and deploy. Second, it is not as well-suited for all tasks. For example, it is not as effective at object detection as some other convolutional neural networks.

Overall, EfficientNet is a powerful convolutional neural network architecture that offers a number of benefits over other convolutional neural networks. It is more efficient, more accurate, and more scalable. However, it is also more complex and not as well-suited for all tasks.

4.3.4 InceptionNet:

InceptionNet is a convolutional neural network (CNN) [17] architecture developed by Google for image classification and object detection. It was first introduced in the paper "Going Deeper with Convolutions" by Szegedy et al. in 2014. InceptionNet won the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC), achieving a top-5 error rate of 6.7%, which was significantly lower than the previous state-of-the-art of 11.7%.

InceptionNet is based on the idea of using inception modules, which are blocks of layers that are designed to learn a combination of local and global features from the input data. Inception modules are composed of smaller convolutional and pooling layers, which are combined to allow the network to learn spatial and temporal features from the input data.

One of the key benefits of InceptionNet is that it is able to train more efficiently and faster than other deep CNNs. This is because inception modules are able to reduce the number of parameters required in the network, while still maintaining high accuracy. Additionally, InceptionNet is able to make use of parallel processing, which further speeds up the training process.

Another benefit of InceptionNet is that it is able to achieve high accuracy on a variety of image classification and object detection tasks. This is because inception modules are able to learn a wide range of features from the input data, which makes the network more robust to changes in the input data.

However, InceptionNet also has some limitations. One limitation is that it can be computationally expensive to train and deploy. This is because InceptionNet is a deep network with a large number of parameters. Additionally, InceptionNet can be sensitive to the size and quality of the training data. If the training data is too small or of poor quality, the network may not be able to learn the necessary features to achieve high accuracy.

Overall, InceptionNet is a powerful and versatile CNN architecture that has achieved state-of-the-art results on a variety of image classification and object detection tasks. While it can be computationally expensive to train and deploy, InceptionNet is a good choice for applications where high accuracy is required.

4.3.5 MobileNet:

MobileNet [18] is a class of convolutional neural network (CNN) that is designed to be efficient and fast on mobile and embedded devices with limited computational resources. It achieves this by using depthwise separable convolutions, which significantly reduce the number of parameters and operations required to process an image. The MobileNet architecture consists of two main types of layers: depthwise convolution layers and pointwise convolution layers.

Depthwise convolution layers apply a single filter to each input channel, resulting in a significant reduction in the number of parameters required. For example, a standard convolution layer with 32 filters applied to a 3-channel image would require 32 * 3 * 3 = 288 parameters. A depthwise convolution layer with 32 filters applied to the same image would only require 32 * 1 * 1 = 32 parameters.

Pointwise convolution layers apply a 1x1 convolution to each input channel, which is used to combine the outputs of the depthwise convolution layers. Pointwise convolution layers have a much smaller number of parameters than standard convolution layers, but they can still learn complex relationships between the input channels.

The MobileNet architecture has been shown to be very effective for a variety of image classification tasks, including object detection, image segmentation, and facial recognition. It has also been used to develop real-time applications, such as player detection in sports applications on embedded devices.

4.3.6 NASNet:

NASNet [19] is a convolutional neural network (CNN) architecture that was developed using neural architecture search (NAS). NAS is a machine learning technique that is used to automatically design neural networks. NASNet was designed to be efficient and accurate, and it has achieved state-of-the-art results on a number of image classification benchmarks.

NASNet is composed of a number of different layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers are used to extract features from the input images. The pooling layers are used to reduce the dimensionality of the feature maps. The fully connected layers are used to classify the images.

NASNet has a number of benefits over other CNN architectures. First, NASNet is more efficient than other CNN architectures. This is because NASNet uses a smaller number of parameters and FLOPs. Second, NASNet is more accurate than other CNN architectures. This is because NASNet was designed using NAS, which is a machine learning technique that is used to automatically design neural networks.

However, NASNet also has some limitations. First, NASNet is more difficult to train than other CNN architectures. This is because NASNet uses a larger number of layers and a more complex architecture. Second, NASNet is more computationally expensive than other CNN architectures. This is because NASNet uses a larger number of parameters and FLOPs.

Overall, NASNet is a powerful and efficient CNN architecture that has achieved state-of-the-art results on a number of image classification benchmarks. However, NASNet is more difficult to train and more computationally expensive than other CNN architectures.

4.3.7 ResNet:

ResNets are a type of convolutional neural network (CNN) [20] that was developed in 2015 for image recognition. They are composed of several residual blocks, each of which contains a convolutional layer, a batch normalization layer, and a ReLU activation function. The output of each residual block is added to the input of the block, which allows the network to learn residual functions with reference to the layer inputs. This makes it easier for the network to train and prevents the vanishing gradient problem.

ResNets have several benefits over other CNN architectures. First, they are more accurate. ResNets won the ImageNet Large Scale Visual Recognition Challenge in 2015, and they have since been used to achieve state-of-the-art results on a variety of other computer vision tasks, including object detection, facial recognition, and image segmentation. Second, ResNets are more efficient. They can train and run faster than other CNN architectures, which makes them more suitable for real-time applications. Third, ResNets are more robust to noise and overfitting. They are able to learn from smaller datasets and generalize better to new data.

However, ResNets also have some limitations. First, they can be more complex and difficult to train than other CNN architectures. Second, they can be more computationally expensive to run. Third, ResNets are not always the best choice for all computer vision tasks. For example, they may not be as effective for tasks that require a high degree of spatial invariance, such as optical flow estimation.

Overall, ResNets are a powerful and versatile CNN architecture that has been used to achieve state-of-the-art results on a variety of computer vision tasks. They are more accurate, efficient, and robust than other CNN architectures, but they can also be more complex and computationally expensive.

ResNets are a powerful and versatile CNN architecture that offers improved accuracy, efficiency, and robustness at the cost of increased complexity and computational expense.

4.3.8 VGGNet:

VGGNet is a convolutional neural network (CNN) [21] model that was proposed by K. Simonyan and A. Zisserman from Oxford University in their 2014 paper, Very Deep Convolutional Networks for Large-Scale Image Recognition. VGGNet is a relatively simple model, with a uniform architecture that consists of a stack of convolutional layers followed by a few fully connected layers. The convolutional layers use small receptive fields (3x3 pixels) and a stride of 1, and the pooling layers use a 2x2 max pooling operation with a stride of 2. This architecture allows VGGNet to learn deep representations of images, while keeping the number of parameters relatively low.

VGGNet has been shown to achieve state-of-the-art results on a variety of image classification tasks, including the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). In ILSVRC 2014, VGGNet won the first place in the localization task and the second place in the classification task. VGGNet has also been successfully applied to other tasks, such as object detection, segmentation, and pose estimation.

One of the main benefits of VGGNet is its simplicity. The uniform architecture of VGGNet makes it easy to understand and implement. Additionally, VGGNet is relatively efficient to train, even on large datasets.

However, VGGNet also has some limitations. One limitation is that VGGNet is a relatively large model, with a large number of parameters. This can make VGGNet slow to run on mobile devices and embedded systems. Additionally, VGGNet is not as robust to noise and occlusions as some other CNN models.

Overall, VGGNet is a powerful and versatile CNN model that has been shown to achieve state-of-the-art results on a variety of image classification tasks. VGGNet is also relatively simple to understand and implement, making it a good choice for researchers and practitioners who are new to CNNs.

4.4 Model Evaluation and Selection

Our evaluation of various models indicated MobileNet offered the best starting point for our image classification task. To maximize accuracy, we then fine-tuned the MobileNet architecture, specifically focusing on its later layers to better recognize the nuances within our dataset. This approach allowed us to leverage MobileNet's efficient design while achieving an even higher degree of precision.

4.5 Results

We evaluated eight pre-trained architectures including AlexNet, DenseNet, EfficientNet, InceptionNet, MobileNet, NASNet, ResNet, and VGGNet.

In the pursuit of optimizing our model's performance, we initially developed a custom Convolutional Neural Network (CNN) architecture, which yielded a Training Accuracy of

76.41% and a Testing Accuracy of 60.33%. Recognizing the potential benefits of leveraging pre-trained models, we experimented with eight established architectures.

Table 1 Shows validation accuracy and training accuracy of various pre-trained model.

S.No.	Model	Testing Accuracy	Training Accuracy
1	AlexNet	51.11	59.03
2	DenseNet	16.67	16.67
3	EfficientNet	64.49	81.61
4	InceptionNet	66.72	83.49
5	**MobileNet**	**67.01**	**86.82**
6	NASNet	60.86	83.11
7	ResNet	25.75	33.41
8	VGGNet	59.14	79.95

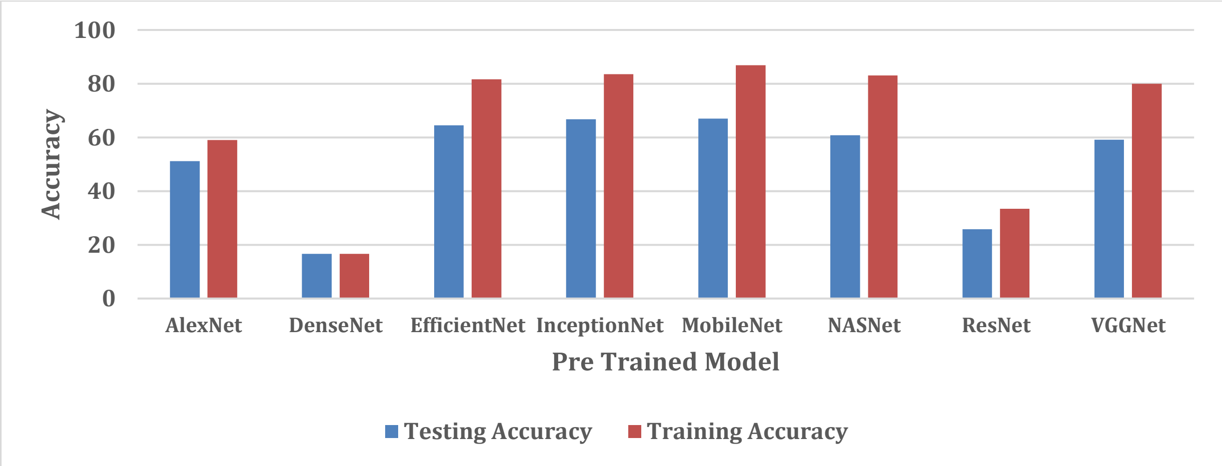

Fig.8 Pretrained Model and its Testing and Training Accuracy

Among these, MobileNet emerged as the most promising, demonstrating superior performance. Subsequently, we undertook fine-tuning processes for both our CNN and MobileNet models. Through meticulous fine-tuning efforts, our CNN model achieved a remarkable testing accuracy of 84% and training accuracy of 89%, solidifying its status as a robust and effective solution for our objectives. These findings underscore the importance of model selection and fine-tuning strategies in enhancing performance outcomes in the domain of convolutional neural networks.

MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Resource Per Layer Type

Type	Mult-Adds	Parameters
Conv 1×1	94.86%	74.59%
Conv DW 3×3	3.06%	1.06%
Conv 3×3	1.19%	0.02%
Fully Connected	0.18%	24.33%

Fig. 9 Model Architecture – MobileNet

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
batch_normalization (BatchNormalization)	(None, 148, 148, 32)	128
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 72, 72, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 34, 34, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
conv2d_3 (Conv2D)	(None, 15, 15, 256)	295,168
batch_normalization_3 (BatchNormalization)	(None, 15, 15, 256)	1,024
max_pooling2d_3 (MaxPooling2D)	(None, 7, 7, 256)	0
conv2d_4 (Conv2D)	(None, 5, 5, 512)	1,180,160
batch_normalization_4 (BatchNormalization)	(None, 5, 5, 512)	2,048
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1,049,088
batch_normalization_5 (BatchNormalization)	(None, 512)	2,048
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 256)	131,328
batch_normalization_6 (BatchNormalization)	(None, 256)	1,024
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 6)	1,542

Total params: 8,265,684 (31.53 MB)
Trainable params: 2,754,054 (10.51 MB)
Non-trainable params: 3,520 (13.75 KB)
Optimizer params: 5,508,110 (21.01 MB)

Fig. 10 Model Architecture - CNN

Fig. 11 Accuracy Graph - CNN

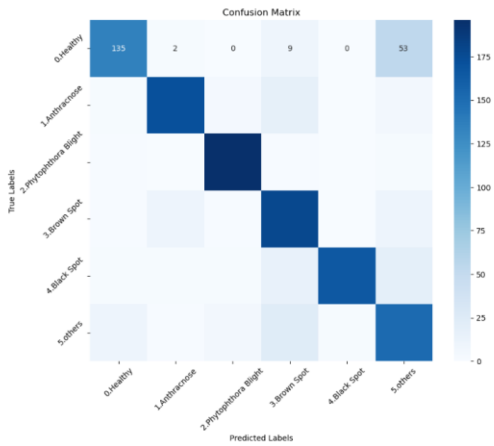

Fig. 12 Confusion Matrix

CHAPTER 5

CONCLUSION

5.1 Conclusion

In conclusion, this work investigated deep learning techniques for papaya fruit disease classification. We first collected a comprehensive dataset and then employed a Convolutional Neural Network (CNN) architecture. To improve performance, we fine-tuned eight pre-trained models and our proposed CNN. The fine-tuned CNN achieved the highest testing accuracy of 84% and training accuracy 89% in classifying papaya fruit diseases. This approach demonstrates the potential of deep learning for robust papaya disease detection. Future work could explore incorporating additional diseases or investigating alternative deep learning architectures for even better accuracy. The success of this project not only highlights the capabilities of deep learning in agriculture but also emphasizes the importance of utilizing a combination of segmentation and classification techniques for improved accuracy and reliability. Moving forward, the insights gained from this project can serve as a foundation for the development of scalable and efficient tools for automated papaya disease classification, contributing significantly to the agricultural sector's efforts in crop preservation and yield optimization.

5.2 Future Work

To enhance and expand the project's capabilities, the dataset will be broadened to incorporate a wider spectrum of papaya diseases and diverse environmental conditions. This will bolster the model's ability to generalize and enhance its adaptability to varying environmental factors. Simultaneously, incorporating advanced algorithms and exploring more deep learning models will open up avenues for exploring alternative approaches to disease detection and classification, while transitioning towards real-time detection and developing a user-friendly mobile application will enable farmers to promptly assess disease presence on-site using their smartphones. Additionally, continuous monitoring, regular updates with new data, and collaboration with agricultural experts will be paramount for maintaining the model's relevance in constantly evolving agricultural landscapes. Moreover, delving into transfer learning and customizing the model for specific regions will further enhance its accuracy and applicability. Throughout the iterative improvement process, user feedback, particularly from farmers, will guide the model's refinement, ensuring practicality and relevance. Ultimately, these future directions aim to transform the project into a versatile and robust tool, making a significant contribution to effective papaya disease detection and classification and crop management in diverse agricultural settings.

REFERENCES

[1] Parle, Milind & Bhardwaj, Gurditta. (2011). Basketful benefits of papaya. International Research Journal of Pharmacy. 2. 6-12.

[2] Verma, Navneet. (2014). A Brief Study on Carica Papaya- A Review. International Journal of Current Trends in Pharmaceutical Research. Vol. 2. 541-550.

[3] Liberato, José. (2012). Papaya anthracnose (Colletotrichum gloeosporioides).

[4] Nelson, Scot. (2008). Phytophthora Blight of Papaya.

[5] Liberato, José & McTaggart, Alistair. (2012). Corynespora brown spot of papaya (Corynespora cassiicola).

[6] Shetty, Swathi & Kumar, V.B. & Shivalli Boregowda, Yogananda & Kumar, N. Kiran. (2021). A review of papaya black spot- a fungal disease (Asperisporium caricae). 55. 1-11.

[7] de Moraes JL, de Oliveira Neto J, Badue C, Oliveira-Santos T, de Souza AF. Yolo-Papaya: A Papaya Fruit Disease Detector and Classifier Using CNNs and Convolutional Block Attention Modules. *Electronics*. 2023; 12(10):2202. https://doi.org/10.3390/electronics12102202.

[8] Md. Tarek Habib, Anup Majumder, A.Z.M. Jakaria, Morium Akter, Mohammad Shorif Uddin, Farruk Ahmed,Machine vision based papaya disease recognition,Journal of King Saud University - Computer and Information Sciences,Volume 32, Issue 3,2020,Pages 300-309.

[9] Ngugi HN, Ezugwu AE, Akinyelu AA, Abualigah L. Revolutionizing crop disease detection with computational deep learning: a comprehensive review. Environ Monit Assess. 2024 Feb 24;196(3):302. doi: 10.1007/s10661-024-12454-z. PMID: 38401024; PMCID: PMC10894121.

[10] Hossen, S.; Haque, I.; Islam, S.; Ahmed, T.; Nime, J.; Islam, A. Deep learning based classification of papaya disease recognition. In Proceedings of the 3rd International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 3–5 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 945–951.

[11] Islam, Md & Islam, Md. Shahriar & Hossen, Md & Emon, Minhaz & Keya, Maria & Habib, Ahsan. (2020). Machine Learning based Image Classification of Papaya Disease Recognition. 1353-1360. 10.1109/ICECA49313.2020.9297570.

[12] Dhanaraju M, Chenniappan P, Ramalingam K, Pazhanivelan S, Kaliaperumal R. Smart Farming: Internet of Things (IoT)-Based Sustainable Agriculture. *Agriculture*. 2022; 12(10):1745. https://doi.org/10.3390/agriculture12101745

[13] Sainath Chaithanya A, Rachana M. (2023) Identification of Diseased Papaya Leaf through Transfer Learning. *Indian Journal of Science and Technology*. 16(48):4676-4687. https://doi.org/10.17485/IJST/v16i48.2690

[14] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E,Advances in Neural Information Processing Systems, ImageNet Classification with Deep Convolutional Neural Networks, volume = 25, year = 2012.

[15] Huang, Gao & Liu, Zhuang & van der Maaten, Laurens & Weinberger, Kilian. (2017). Densely Connected Convolutional Networks. 10.1109/CVPR.2017.243.

[16] Tan, Mingxing & Le, Quoc. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.

[17] Pandit, Tejas & Kapoor, Akshay & Shah, Rishi & Bhuva, Rushi. (2020). Understanding Inception Network Architecture For Image Classification. 10.13140/RG.2.2.16212.35204.

[18] Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[19] Zoph, Barret & Vasudevan, Vijay & Shlens, Jonathon & Le, Quoc. (2018). Learning Transferable Architectures for Scalable Image Recognition. 8697-8710. 10.1109/CVPR.2018.00907.

[20] Liang, Jiazhi. (2020). Image classification based on RESNET. Journal of Physics: Conference Series. 1634. 012110. 10.1088/1742-6596/1634/1/012110.

[21] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556