PLANT LEAF DISEASE DETECTION USING DEEP LEARNING

A PROJECT REPORT

Submitted by

JAYANTHINI R (810020205026)

JEEVA K (810020205027)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

UNIVERSITY COLLEGE OF ENGINEERING

BIT CAMPUS, TIRUCHIRAPPALLI

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2024

UNIVERSITY COLLEGE OF ENGINEERING
BIT CAMPUS
TIRUCHIRAPPALLI-620 024

BONAFIDE CERTIFICATE

Certified that this project report "**PLANT LEAF DISEASE DETECTION USING DEEP LEARNING"** is the bonafide work of "**MISS.JAYANTHINI R (810020205026)** and **MISS. JEEVA K(810020205027)"**who carried out the project work under my supervision.

SIGNATURE
Dr. G. ANNAPOORANI
HEAD OF THE DEPARTMENT
Assistant Professor
Department of IT/CSE
University College of Engineering,
Anna University-BIT Campus,
Tiruchirappalli-620 024

SIGNATURE
Dr. R. KAVITHA
SUPERVISOR
Assistant Professor
Department of CSE
University College of Engineering,
Anna University-BIT Campus,
Tiruchirappalli-620 024

Submitted for the project Viva - voce examination held on …….....

Internal Examiner **External Examiner**

DECLARATION

We hereby declare that the work entitled "**PLANT LEAF DISEASE DETECTION USING DEEP LEARNING**" is submitted in partial fulfillment of the requirement for the award of the degree in B.Tech, in University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. It is the record of our own work carried out during the academic year 2023 – 2024 under the supervision and guidance of **Dr.R.KAVITHA**, Assistant Professor, Department of CSE, University College of Engineering, BIT Campus, Anna University, Tiruchirappalli. The extent and source of information are derived from the existing literature and have been indicated through the dissertation at the appropriate places.

R. JAYANTHINI (810020205026)

K JEEVA (810020205027)

I certify that the declaration made above by the candidates is true

Signature of the Guide

Dr.R.KAVITHA,
Assistant Professor,
Department of CSE,
University College of Engineering,
Anna University – BIT Campus,
Tiruchirappalli – 620 024.

ACKNOWLEDGEMENT

We would like to thank our honorable Dean **Dr.T.SENTHIL KUMAR**, Professor for having provided us with all required facilities to complete our project without hurdles.

We would also like to express our sincere thanks to **Dr.G.ANNAPOORANI**, Head of the Department, Department of Computer Science and Engineering, further valuable guidance, suggestions and constant encouragement paved way for the successful completion of this project work.

We would like to thank our Project Coordinator **Dr. S. USHA**, Assistant Professor, Department of Computer Science and Engineering for her kind support.

We would like to thank and express our deep sense of gratitude to our project guide **Dr.R.KAVITHA,** Assistant Professor, Department of Computer Science and Engineering, for her valuable guidance throughout the project.

We also extend our thanks to all other teaching and non-teaching staff for their encouragement and support.

We thank our beloved parents and friends for their full support in the moral development of this project.

ABSTRACT

Economy contribute the most of the productivity of agriculture. In agricultural field, the disease in plants is more common and the detection of disease in plants has become more feasible. To achieve this goal, researchers have been used many deep learning methods like CNN, Mobilenet v2, YOLO V3, Densenet to detect the disease and to cure them. This novel uses a technique called MobileNet and the datasets are Basic Indian Plant Leaf Images to handle this problem. MobileNet is specifically optimized for resource-constrained devices, offering a good balance between model size and accuracy. Diseases in plants from agricultural field have become common today and detection of diseases in plants have become feasible. There have been several versions and variations of MobileNet, each offering different trade-offs between model size, speed, and accuracy to cater to different use cases and deployment scenarios.

Keywords: CNN, Resnet, YOLO V3.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	-	Artificial Intelligence
DL	–	Deep Learning
ML	–	Machine Learning
CNN	–	Convolutional Neural Network
Et.al.,	–	And others
SGD	–	Stochastic Gradient Descent.
ReLU	–	Rectified Linear Unit
DRDN	–	Deep Residual Dense Network
DNN	-	Deep Neural Network
CSV	-	Comma Separated Values
CPU	-	Central Processing Unit
GPU	-	Graphical Processing Unit
TPU	-	Tensor Processing Unit
PIL	-	Pillow Library
ONNX	-	Open neural network exchange
VGG	-	Visual Geometric Group
Resnet	-	Residual Network

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Plant leaf disease detection can be done using machine vision equipment to accure images and evaluate for diseases and pests. This process involves a series of steps including image acquisition, image preprocessing, segmentation, feature extraction, selection and detection and classification.

In recent years, agriculture has encountered unprecedented challenges due to various factors such as climate change, globalization, and population growth. Among these challenges, the detrimental impact of plant diseases on crop yield and quality stands as a critical concern. Plant diseases not only compromise agricultural productivity but also threaten food security worldwide. Timely and accurate detection of these diseases is imperative for effective disease management and sustainable agriculture practices.

Traditional methods of disease diagnosis often rely on visual inspection by trained agronomists, which can be subjective, time-consuming, and prone to errors. However, the advent of technology, particularly in the fields of computer vision and machine learning, offers promising solutions to address these limitations. Automated systems for plant leaf disease detection, based on image processing and artificial intelligence techniques, have gained significant attention in recent years for their potential to revolutionize crop disease management.

1.1.1 INTRODUCTION TO LEAF DISEASE DETECTION

Leaf disease detection is important because profit and loss depend on production. So that here use deep learning techniques to detect apple, grape, corn, potato, and tomato plant leaves diseases. That contains 38 different classes, twenty-four types of leaf diseases and twenty-four thousand leaves images are used. Apple, grape, corn, potato, and tomato plant leaves which are categorized total 24 types of labels apple label namely: Apple scab, Black rot, apple rust, and healthy. Grape label namely: Black rot, Esca, healthy, and Leaf blight. Corn label namely: Corn Cercospora spot Gray, Corn rust, Corn healthy, Corn Northern Blight. Potato label namely: Early blight, healthy, and Late blight. Tomato label namely: bacterial spot, early blight, healthy, late blight, leaf mold, leaf spot, spider mite, target sport, mosaic virus.

The dataset consist of 31,119 images of apple, grape, potato and tomato, all Images are resized into 256 x 256, that images divided into two parts training and testing dataset

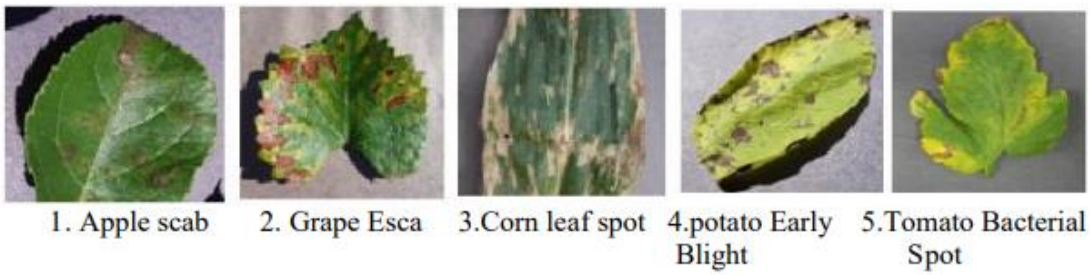

Fig.1.1.1.1 Leaves with Disease part [11]

In fig.1.1 we can see vegetable and fruit leaves like potato, tomato, corn, apple, grape with diseased part this disease can be easily detected using deep learning techniques [13].

This disease detected using convolutional neural network (CNN), and also this model is compared with VGG16. Images are resized into 224 x 224[13]

1.2 DEEP LEARNING

Deep learning is a method in artificial intelligence (AI) that teaches computers to process data in a way that is inspired by the human brain. Deep learning is a type of machine learning and artificial intelligence(AI) that imitates the way humans gain certain types of knowledge. It is also used to automate tasks that would normally need human intelligence, such as describing images. Deep learning is an important element of data science, including statistics and predictive modeling. It is extremely beneficial to data scientists who are tasked with collecting, analyzing and interpreting large amounts of data; deep learning makes this process faster and easier.

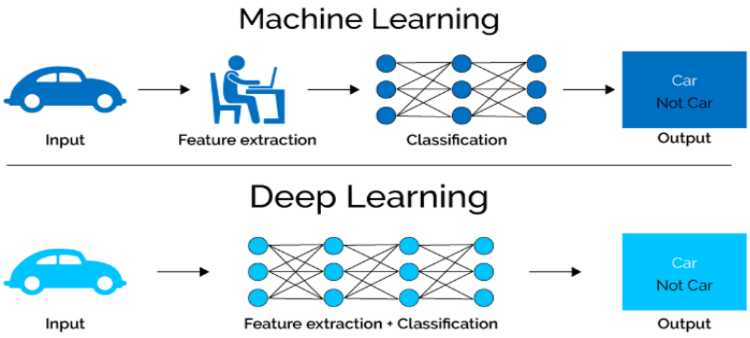

Fig.1.2.1 Machine Learning vs Deep Learning

Deep Learning are applied in Computer vision, image processing, automated driven and signal preprocessing. Deep learning model architecture is crucial for the success of the plant leaf disease detection task. Convolutional Neural Networks (CNNs) are widely used for image classification tasks due to their ability to automatically learn hierarchical features from raw pixel data.

1.2.1 Deep Learning Algorithm

Convolutional Neural Networks (CNNs) are widely employed due to their effectiveness in image classification tasks. Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing and analyzing visual data, such as images. CNNs are highly effective in tasks like image classification, object detection, and segmentation due to their ability to automatically learn hierarchical patterns and features from raw pixel data.

MobileNet is a type of Convolutional Neural Network (CNN) architecture specifically designed for efficient inference on mobile and embedded devices with limited computational resources. The key innovation of MobileNet lies in its depth-wise separable convolutions, which drastically reduce the number of parameters and computations compared to traditional convolutional layers while still preserving accuracy.

1.3 OBJECTIVE

- The objective of this research is to concentrate based on potato, tomato, corn, grapes, and apple leaf disease detection using CNN.

- To develop reliable and efficient methods for identifying and diagnosing diseases affecting plant leaves.

- To empower farmers and agricultural stakeholders with the tools and knowledge needed to protect crops, ensure food security, and sustainably manage agricultural systems in the face of plant diseases and other challenges.

- CNN is used for examine the healthy and diseased plants leaves.

CHAPTER 2
LITERATURE SURVEY

Emmanul Moupojou et.al., proposed a “A Dataset of Field Plant Images for Plant Disease Detection and Classification with Deep Learning” in the year 2023. The paper focuses on creating the dataset and outlining the significance of plant disease detection for agriculture. It may review existing literature on deep learning-based approaches to plant disease detection and emphasize the need for high-quality datasets to advance research in this area. The introduction may also provide an overview of the dataset creation process and the methods used to collect and label field plant images. The procedures used to collect images of plant leaves affected by various diseases from field environments. It may describe the selection of plant species, disease types, and geographical locations to ensure diversity and representativeness in the dataset. This section outlines the experimental setup for training and evaluating deep learning models using the dataset. It describes the architecture of the neural network models employed, hyper parameters chosen for training, and evaluation metrics used to assess model performance. Details about data preprocessing, model training procedures, and model evaluation protocols are provided. The results section presents the performance of deep learning models trained on the dataset for plant disease detection and classification tasks. It includes quantitative metrics such as accuracy, precision, recall, and F1 score, as well as qualitative analysis of model predictions. The discussion interprets the results, highlights strengths and limitations of the dataset and models, and suggests areas for future research and improvement. The key findings of the study and emphasizes the importance of the dataset for advancing research in plant disease detection with deep learning. It may also discuss potential applications of the dataset in agriculture, such as developing decision support tools for farmers or guiding breeding programs for disease-resistant crop varieties.

Shujuan Zhang et.al., proposed a “Plant Disease Detection and Classification by Deep learning” in the year of 2021. The primary objective of the study was to develop an accurate and efficient method for detecting and classifying plant diseases using deep learning algorithms. The authors aimed to address the challenges associated with traditional methods of disease diagnosis in agriculture by leveraging the capabilities of deep learning models. The authors employed deep learning techniques, specifically Convolutional Neural Networks (CNNs), which are well-suited for image classification tasks. They collected a dataset of plant images containing examples of healthy plants as well as plants affected by various diseases. The dataset was carefully curated and annotated with disease labels to facilitate model training. The CNN architecture used in the study was designed to automatically learn discriminative features from the input images, enabling the model to accurately classify the presence and type of disease in plant leaves. The training process involved optimizing the network's parameters using gradient-based optimization algorithms, such as stochastic gradient descent (SGD) or Adam, to minimize classification errors. The proposed deep learning model demonstrated promising results in detecting and classifying plant diseases. The model achieved high accuracy rates in distinguishing between healthy and diseased plants and accurately identifying the specific disease types. The performance of the model was evaluated using standard metrics such as accuracy, precision, recall, and F1 score, which indicated its effectiveness in disease diagnosis. The authors discussed the significance of their findings in the context of agricultural applications, emphasizing the potential impact of deep learning-based disease detection systems on crop yield and food security. They highlighted the advantages of using deep learning models, such as their ability to generalize well to unseen data and their scalability for large-scale deployment in agricultural settings.

Hassan Amin et.al., proposed a “End-to-End Deep learning model for Corn leaf Disease Classification” in the year of 2022. This research addresses the critical issue of plant diseases, which pose a significant threat to global food security. Identifying plant diseases accurately and swiftly is essential for preventing crop losses and ensuring healthy agricultural production. They collected a dataset of corn leaf images containing examples of both healthy leaves and leaves affected by different diseases, such as rust, blight, and leaf spot. The deep learning model architecture utilized in the study was tailored specifically for the task of corn leaf disease classification. It consisted of multiple layers of convolutional neural networks (CNNs) followed by fully connected layers for classification. The model was trained using a large dataset of labeled images, and techniques such as data augmentation and transfer learning were employed to improve its performance. The proposed end-to-end deep learning model demonstrated impressive results in accurately classifying corn leaf diseases. It achieved high levels of accuracy in distinguishing between healthy and diseased leaves and effectively identifying the specific types of diseases present. The performance of the model was evaluated using standard evaluation metrics such as accuracy, precision, recall, and F1 score, which indicated its robustness and effectiveness in disease classification. They highlighted the advantages of the end-to-end model architecture, including its simplicity, efficiency, and scalability for real-world applications in agricultural settings. Their study contributes to the advancement of deep learning techniques in agriculture and provides a valuable tool for farmers and researchers in combating corn leaf diseases. Overall, the paper underscores the importance of leveraging artificial intelligence and machine learning technologies to address challenges in crop health monitoring and disease management, ultimately contributing to sustainable agriculture and food security.

Sunil C.K et.al., proposed a “Cardamon Plant Disease Detection Approach using Efficient NetV2 in the year of 2022. Introducing a novel methodology for accurately detecting diseases affecting cardamom plants by leveraging the EfficientNetV2 architecture. The primary objective of the study was to develop an efficient and accurate approach for detecting diseases in cardamom plants using deep learning techniques. Cardamom is an economically important spice crop, and early detection of diseases is crucial for minimizing crop losses and ensuring sustainable production. The authors aimed to address this challenge by proposing a deep learning-based solution tailored specifically for cardamom plant disease detection. They collected a dataset of cardamom plant images containing examples of both healthy plants and plants affected by various diseases common to cardamom cultivation. The deep learning model architecture employed in the study consisted of EfficientNetV2 pre-trained on a large-scale image dataset followed by additional layers for disease classification. Transfer learning techniques were utilized to fine-tune the pre-trained model on the cardamom plant dataset, thereby leveraging existing knowledge from the broader image domain to improve performance on the specific task of disease detection. The proposed approach demonstrated promising results in accurately detecting and classifying diseases in cardamom plants. The model achieved high levels of accuracy in distinguishing between healthy and diseased plants and effectively identifying the specific types of diseases present. Their study contributes to the advancement of deep learning techniques in agriculture and provides a valuable tool for cardamom farmers and researchers in combating plant diseases. Overall, the paper underscores the importance of harnessing advanced technologies for disease detection and management in agriculture, ultimately promoting sustainable crop production and food security.

Changjian Zhou et.al., proposed by “Tomato Leaf disease Identification by Restructured Deep Residual Dense Network” in the year of 2021. The main objective of the study was to develop an effective and reliable method for identifying diseases in tomato leaves using deep learning techniques. Tomato is one of the most economically important crops worldwide, but it is susceptible to various diseases that can cause significant yield losses. The authors aimed to address this challenge by proposing an advanced deep learning-based approach tailored specifically for tomato leaf disease identification. The DRDN model combines the strengths of residual connections and dense connections, which are well-known techniques in deep learning for improving model performance and training stability. The model architecture was carefully designed to effectively capture and leverage hierarchical features in tomato leaf images. A dataset of tomato leaf images containing examples of healthy leaves as well as leaves affected by various diseases, such as early blight, late blight, and bacterial spot, was collected for training and evaluation. The deep learning model was trained using this dataset, with techniques such as data augmentation and transfer learning employed to enhance model generalization and performance. The proposed approach demonstrated impressive results in accurately identifying diseases in tomato leaves. The DRDN model achieved high levels of accuracy in distinguishing between healthy and diseased leaves and effectively classifying the specific types of diseases are available. They highlighted the potential of advanced deep learning techniques, such as the restructured DRDN model, for improving disease management practices and supporting decision-making in agricultural settings. The importance of leveraging advanced technologies for disease detection and management in agriculture, ultimately promoting sustainable crop production and food security.

S K Mahmudul Hassan et.al., proposed a “Plant Disease Identification using a Novel Covolution Neural Network” in the year of 2021. The introduction of a novel convolutional neural network (CNN) technique for plant disease identification represents a groundbreaking advancement in agricultural technology. This innovative approach harnesses the power of deep learning, specifically CNNs, renowned for their proficiency in recognizing intricate patterns in image data. By training on extensive datasets of plant images, the CNN model learns to accurately detect and classify various diseases based on visual symptoms, offering farmers a rapid and reliable tool for diagnosing crop ailments. The strength of this novel CNN methodology lies in its ability to efficiently process vast amounts of image data, enabling swift and precise disease identification. Leveraging the hierarchical layers of convolution and pooling operations, the model extracts increasingly abstract features from the input images, effectively capturing the subtle visual cues indicative of different diseases. Through iterative training on diverse datasets encompassing a wide range of crops and diseases, the CNN learns to generalize its understanding, achieving robust performance across various agricultural contexts. Furthermore, the accessibility and scalability of this CNN methodology make it particularly well-suited for deployment in agricultural settings worldwide. With advancements in imaging technology and the proliferation of mobile devices, farmers can easily capture and upload images of their crops for analysis using the CNN model. This democratization of diagnostic tools empowers farmers, especially those in remote or underserved regions, to make informed decisions about disease management and crop protection strategies. In addition to its practical applications in crop health monitoring, the CNN-based approach holds promise for advancing our understanding of plant-pathogen interactions. By analyzing the features learned by the model, researchers can gain insights into the underlying biological mechanisms of disease development and identify novel targets for disease control measures.

Asad Khatlak et.al., proposed a “Automatic Detection of Citrus Fruit and leaves Diseases using Deep Neural Network Model” in the year of 2021. The automatic detection of citrus fruit and leaf diseases using a deep neural network (DNN) model represents a significant advancement in agricultural technology, offering a powerful tool for early disease detection and management in citrus crops. Leveraging the capabilities of deep learning, this innovative approach enables the accurate identification and classification of various diseases affecting citrus fruits and leaves based on visual symptoms. By training on extensive datasets of citrus images depicting healthy and diseased specimens, the DNN model learns to distinguish between different disease types and accurately categorize them, providing farmers with timely insights into the health status of their crops. The strength of this DNN-based methodology lies in its ability to effectively analyse large volumes of image data, extracting meaningful features that characterize citrus diseases. Through multiple layers of neural network architecture, the model processes the input images, progressively refining its understanding of disease patterns and enhancing its diagnostic accuracy. By learning from diverse examples of citrus diseases, the DNN model achieves robust performance across various environmental conditions and disease severities, ensuring reliable disease detection under real-world agricultural settings. Early detection of citrus fruit and leaf diseases is crucial for implementing timely interventions and preventing the spread of infections throughout orchards. By automatically identifying diseased specimens based on visual symptoms such as discoloration, lesions, or deformities, the DNN model enables farmers to take proactive measures, such as targeted pesticide applications, cultural practices, or removal of infected plants. This proactive approach to disease management not only helps minimize crop losses but also reduces the reliance on chemical inputs, promoting sustainable agricultural practices and environmental stewardship.

CHAPTER 3

DATASET DESCRIPTION

The New Plant Diseases Dataset offers a valuable resource for researchers and developers working in the field of plant health classification. This dataset, available on Kaggle, provides a collection of images categorized by various plant diseases. The dataset boasts around 87,000 RGB (Red, Green, Blue) images, capturing both healthy and diseased crop leaves. These images are further classified into 38 distinct categories, allowing for targeted analysis of specific plant ailments. The dataset is conveniently divided into a training and validation set, following an 80/20 split. This ensures a robust training process for machine learning models while reserving a portion of the data for evaluating model performance. While the original source of the images is not explicitly mentioned, the dataset description on Kaggle indicates it was recreated using offline augmentation techniques. This suggests the possibility of the original dataset being derived from various sources and then subjected to image augmentation methods to increase data variety and improve model generalization. Overall, the New Plant Diseases Dataset provides a valuable starting point for researchers and developers working on plant health classification. By acknowledging its limitations and employing appropriate data handling techniques, this dataset can significantly contribute to advancements in automated plant disease detection and improved agricultural practices.

CHAPTER 4

SYSTEM STUDY

4.1 EXISTING SYSTEM

The existing plant leaf disease detection to follow the steps given below

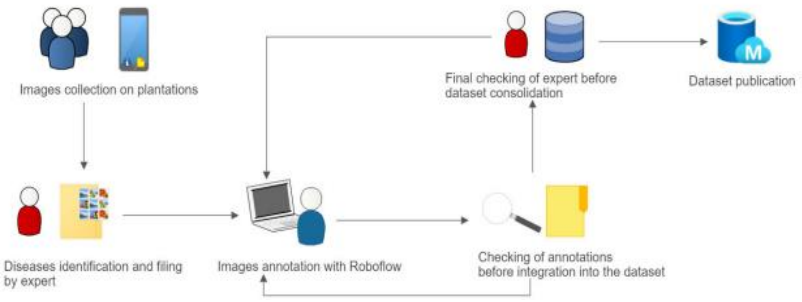

Fig 4.1.1 Field plant dataset setup workflow

Field plant images are collected from various agricultural locations, capturing a wide range of plant species, growth stages, and environmental conditions. The images encompass both healthy plants and plants exhibiting symptoms of different diseases. To enhance the diversity and robustness of the dataset, data augmentation techniques are applied to the collected images. This may include techniques such as rotation, flipping, scaling, and color augmentation to simulate variations in lighting, perspective, and occlusion. Deep learning models, such as Convolutional Neural Networks (CNNs), are trained on the annotated dataset using state-of-the-art training methodologies. The dataset is split into training, validation, and test sets to evaluate the performance of the trained models accurately. These metrics provide insights into the model's ability to accurately detect and classify plant diseases.

To produce a high-quality dataset of field plant images that can serve as a benchmark for plant disease detection and classification research. By leveraging deep learning techniques, the dataset facilitates the development of robust and accurate models capable of detecting a wide range of plant diseases in real-world. By providing researchers and practitioners with access to high-quality data, the system enables advancements in disease management, crop protection, and agricultural sustainability.

4.2 DISADVANTAGES OF EXISTING SYSTEM

- Data Collection Challenges: Training complex CNNs often requires very large datasets to achieve optimal performance. This can be a challenge for agricultural applications where obtaining diverse and extensive data might be difficult. It may require extensive fieldwork, coordination with agricultural experts, and access to various plant species, diseases, and environmental conditions.
- Higher Computational Cost: Traditional CNNs often require more processing power due to their complex architectures. This can make them slow for real-time applications on mobile devices or limit their deployment on devices with lower resources.
- Data Augmentation Limitation: While data augmentation techniques help increase dataset diversity, they may also introduce artificial patterns or artifacts into the data. Overly aggressive augmentation or inappropriate augmentation strategies could potentially degrade model performance or introduce biases.
- Larger Model Size: Complex CNNs can result in larger models that take up more storage space and memory. This can be an issue for mobile devices with limited storage capacity.

CHAPTER 5

SYSTEM REQUIREMENTS

5.1 HARDWARE REQUIREMENTS

System	:	PC OR LAPTOP
Processor	:	Intel(R) Core(TM) i5-7200U CPU / AMD
RAM	:	8 GB
GPU	:	T4 or P100 (Virtual)

5.2 SOFTWARE REQUIREMENTS

OPERATING SYSTEM	:	WINDOWS 11
LANGUAGE USED	:	PYTHON
TOOLS	:	KAGGLE or GOOGLE COLAB

CHAPTER 6

SOFTWARE SPECIFICATIONS

6.1 LIBRARIES USED

- GLOB
- PANDAS
- KERAS
- TENSORFLOW
- PILLOW
- ONNXRUNTIME
- NUMPY
- CURL
- TORCH

GLOB:

The glob library is generally used for finding all files matching a specific pattern within directories. It simplifies tasks like locating all images with a particular extension or finding all files starting with a specific name within a folder structure.

PANDAS:

Pandas is a powerful Python library specifically designed for data manipulation and analysis. It excels at working with tabular data, allowing you to load data from spreadsheets, CSV files, or databases into easy-to-use data structures called DataFrames.

KERAS:

Keras is a high-level library built on top of frameworks like TensorFlow, simplifying deep learning model development. It provides a user-friendly interface for defining neural network architectures, handling data pre-processing, and training models.

TENSORFLOW:

Tensorflow excels at numerical computations and building deep learning models. TensorFlow provides tools for defining neural network architectures, handling large datasets, and efficiently training models on CPUs, GPUs, or specialized hardware like TPUs.

PILLOW:

The Pillow library (often referred to as PIL Fork) is a user-friendly Python library for image processing tasks. It provides functionalities for loading images in various formats (JPEG, PNG, etc.), manipulating them (resizing, cropping, rotating), and saving them in different formats.

ONNXRUNTIME:

ONNX Runtime is a software library that acts as an interpreter for models saved in the Open Neural Network Exchange (ONNX) format. This format provides a standardized way of representing deep learning models, allowing them to be run by various frameworks and platforms.

NUMPY:

NumPy (Numerical Python) is a fundamental library in Python for scientific computing and data analysis. It excels at working with multidimensional arrays, offering efficient operations and functionalities not readily available in standard Python lists.

CURL:

The curl library, often referred to as "curl" itself, is a command-line tool and library used for transferring data over various network protocols. It provides a powerful and versatile way to interact with web servers and download or upload files.

TORCH:

Torch is a powerful deep learning library offering tools to build and train various neural networks. It provides pre-trained models for tasks like image classification, and allows for efficient computations on GPUs or CPUs.

CHAPTER 7

PROJECT DESCRIPTION

7.1 PROPOSED SYSTEM

This project proposes a mobile system for plant leaf disease detection using deep learning. The system leverages a pre-trained convolutional neural network (CNN) like MobileNet v2 to analyze images captured on mobile devices. Users take pictures of plant leaves, and the images are pre-processed on the device for compatibility with the CNN model. This pre-processed image is then fed into the on-device CNN, which analyzes the image features and predicts the presence of a specific disease in real-time. The system displays the predicted disease and confidence level, empowering users with immediate insights into potential plant health issues. This mobile approach enables early disease detection, is accessible in the field for farmers and agricultural professionals. The provided code snippets demonstrate core functionalities: using PyTorch to access the pre-trained model, perform image pre-processing, and export the model for mobile deployment. This system offers a user-friendly and accessible tool for improved crop health management.

7.2 ALGORITHMS USED

The supervised learning algorithm used in this work,

- Convolutional Neural Network

7.2.1 CONVOLUTIONAL NEURAL NETWORK

The architecture of a CNN draws inspiration from the human visual cortex, mimicking the way neurons connect and process visual information. A typical CNN for plant disease detection will have multiple hidden layers stacked

on top of an input layer that receives the pre-processed image data. These hidden layers include convolutional layers that use filters to extract features from the image, followed by pooling layers to reduce dimensionality and activation layers to introduce non-linearity. Finally, fully connected layers process the extracted features and generate probabilities for each disease class, enabling the model to classify the plant disease present in the image.

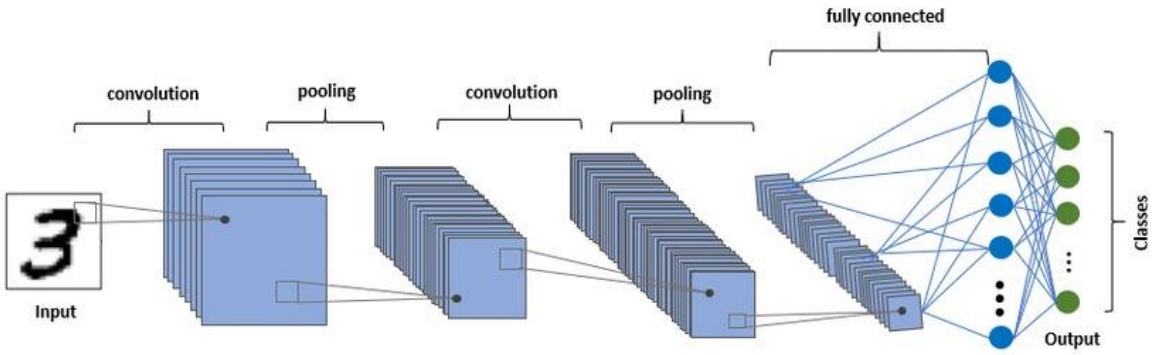

Fig 7.2.1.1 Layers in CNN

This project leverages the power of CNNs to achieve accurate plant disease detection through image analysis. By learning the distinctive visual patterns associated with different diseases, the CNN model can effectively identify the condition based on a given leaf image.

Convolutional layer:

Convolutional layers are the workhorses of plant leaf disease detection in deep learning models. They automatically learn filters that can identify patterns in leaf images, like discolorations, spots, or wilting. By stacking these layers, the model can extract increasingly complex features, ultimately allowing it to differentiate between healthy and diseased leaves.

Pooling Layer:

Pooling layers follow convolutional layers in plant leaf disease detection models. They act like a summarizer, reducing the image size while capturing the

most important features identified by the convolutional layers. This down sampling process makes the model more efficient and helps prevent overfitting on the training data.

Activation Layer:

Activation layers in plant leaf disease detection models play a crucial role in introducing non-linearity. They transform the outputs from convolutional layers, creating a threshold for what information gets passed on. This allows the model to learn complex relationships between the features and ultimately make accurate classifications between healthy and diseased leaves. Common choices include ReLU (Rectified Linear Unit) for its efficiency and ability to handle vanishing gradients.

Flattening:

In plant leaf disease detection using convolutional neural networks (CNNs), the flattening layer plays a critical role in transitioning from spatial data to a format suitable for classification. After convolutional and pooling layers have extracted features, the flattening layer transforms the convolutional output from a multi-dimensional array into a one-dimensional vector. This vector represents all the features from the convolutional layers, preparing the data for feeding into fully connected layers, which make the final disease classification.

7.2.1.1 MOBILENET

MobileNet is a specific convolutional neural network (CNN) architecture designed for **efficiency** on mobile and embedded devices. While powerful CNNs like VGG or ResNet achieve high accuracy, they can be computationally expensive. This becomes a limitation for real-time applications on smartphones or resource-constrained devices in agriculture.

Here's how MobileNet addresses this challenge:

- **Depthwise Separable Convolutions:** MobileNet replaces traditional convolutions with depthwise separable convolutions. These break down the convolution into two steps: a depthwise convolution using lightweight filters for feature extraction, followed by a pointwise convolution to reduce the number of channels. This significantly reduces computational cost while maintaining feature learning capabilities.
- **Linear Bottlenecks:** MobileNet utilizes linear bottlenecks within its architecture. These are thin layers placed between the depthwise and pointwise convolutions. They compress the feature maps before applying the pointwise convolution, further reducing the number of parameters and computations.

By incorporating these techniques, MobileNet offers a **compact and efficient** alternative to traditional CNNs. This makes it particularly suitable for plant leaf disease detection applications on mobile devices, allowing farmers to diagnose plant health in the field using their smartphones.

MobileNet can be used as a pre-trained model, where you leverage the weights learned on a large image dataset for your plant disease classification task. This can be a good starting point for fine-tuning the model on your specific plant disease data.

7.3 ADVANTAGES OF PROPOSED SYSTEM

The proposed system for plant leaf disease detection, leveraging the MobileNet architecture, offers a compelling array of advantages. MobileNet's lightweight and efficient design ensure that the system can be seamlessly integrated into mobile devices, enabling on-the-go disease detection directly in the field. This efficiency not only facilitates real-time analysis but also reduces reliance on constant internet connectivity, making it ideal for agricultural settings where connectivity may be limited. Moreover, the speed of MobileNet

enables rapid inference, allowing for timely intervention and decision-making by farmers. Its scalability further enhances its applicability across farms of varying sizes, catering to the needs of both smallholder farmers and large commercial operations. Despite its lightweight nature, MobileNet maintains competitive accuracy levels, ensuring reliable disease detection results. Coupled with a user-friendly interface, which simplifies the process of capturing and analyzing leaf images, the system becomes an accessible and practical tool for farmers to proactively manage plant health and maximize crop yields.

7.4 SYSTEM ARCHITECTURE

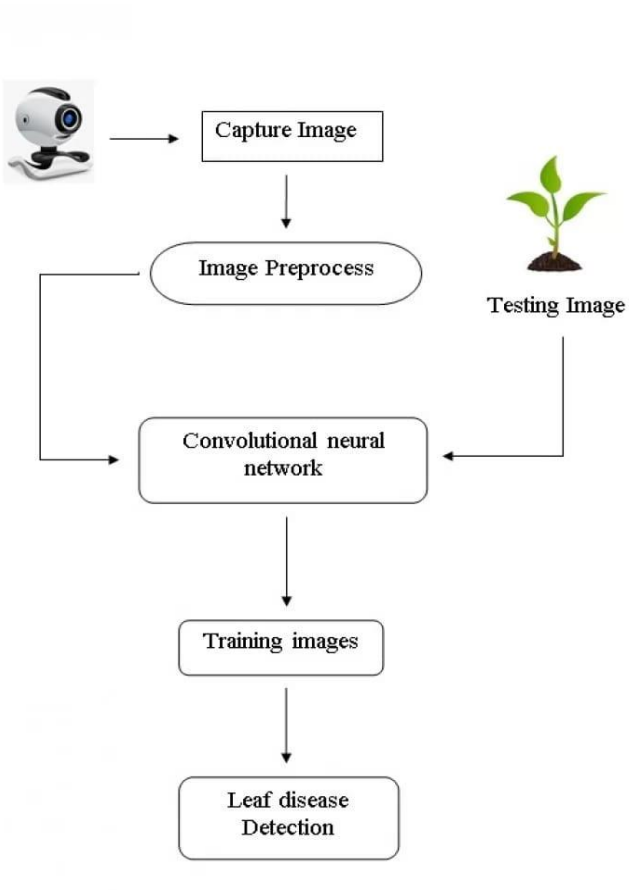

Fig 7.4.1 System Architecture diagram

This diagram depicts a leaf disease detection system powered by a convolutional neural network (CNN). The process starts with capturing a leaf

image. Preprocessing readies the image for the CNN, which has been trained on a vast dataset of labeled healthy and diseased leaves. The CNN's core is a series of convolutional and pooling layers. Convolutional layers extract features from the image, while pooling layers summarize the data and reduce complexity. The preprocessed image is fed through these layers, allowing the CNN to make a disease classification for the captured leaf. This system offers an automated way to identify plant diseases, potentially empowering farmers with early detection and treatment strategies.

CHAPTER 8

SYSTEM IMPLEMENTATION

8.1 MODULE DESCRIPTION:

MODULES:

This project is divided into two modules,

- Preprocessing
- Feature Extraction

8.1.1 Preprocessing:

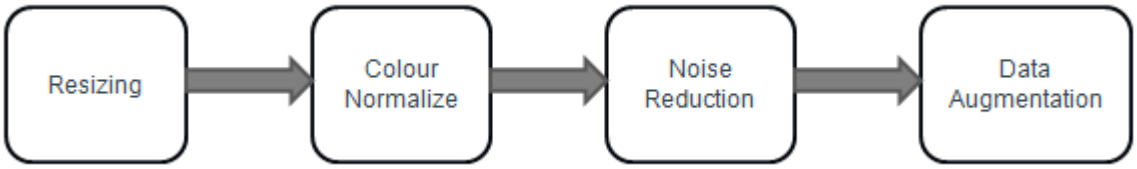

Fig 8.1.1.1 Preprocessing module diagram

This diagram showcases the preprocessing pipeline for a plant leaf disease identification project using a CNN. A raw image goes through a series of steps: resizing for consistency, converting to grayscale for simpler analysis, normalization for efficient training, and potentially background removal and data augmentation to enrich the training data. These steps prepare the image for the CNN, allowing it to focus on disease features within the leaf and ultimately achieve better classification accuracy.

8.1.1.1 Resizing

Resizing in plant leaf disease detection plays a key role. It ensures all images are a consistent size for the CNN model, allowing it to focus on the leaf area and relevant features for disease identification. This standardization also

improves computational efficiency by reducing the number of pixels the model needs to analyze.

8.1.1.2 Color Normalize

Color normalization in your MobileNet project for plant disease detection re-scales the image's pixel values to a common range (often 0-1). This creates a standardized input for MobileNet, regardless of lighting variations, and improves training efficiency by focusing the model on disease patterns within the normalized color data, ultimately leading to better disease classification accuracy.

8.1.1.3 Noise Reduction

Noise reduction is generally less common in plant disease detection with MobileNet for a few reasons. Controlled image capture environments minimize noise, and disease signatures themselves might be misinterpreted as noise by generic algorithms. Additionally, MobileNet prioritizes efficiency, and noise reduction can be computationally expensive. However, noise reduction might be used for images captured in uncontrolled environments or to address specific noise types like salt-and-pepper noise.

8.1.1.4 Data Augmentation

Data augmentation tackles the challenge of limited real-world disease data in your MobileNet project. It artificially creates variations of existing images (rotations, flips, scaling) to enrich the training dataset. This helps MobileNet become more robust to variations in real-world leaf images, reducing overfitting and allowing it to learn generalizable features for accurate disease classification.

8.1.2 Feature Extraction:

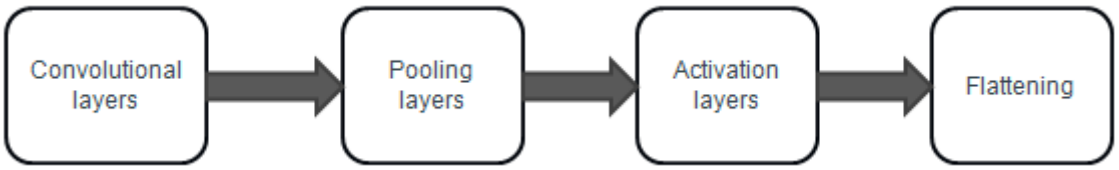

Fig 8.1.2.1 Feature Extraction module diagram

These layers apply filters to the preprocessed image, identifying patterns and extracting features relevant to disease classification. These filters are learned by the MobileNet model during training on a large dataset of leaf images. Pooling layers are often included after convolutional layers to reduce the image size and capture the most important features, promoting efficiency and reducing overfitting. By processing the image through these layers, MobileNet extracts features that differentiate healthy from diseased leaves, ultimately enabling accurate disease classification.

8.2 DATAFLOW DIAGRAM

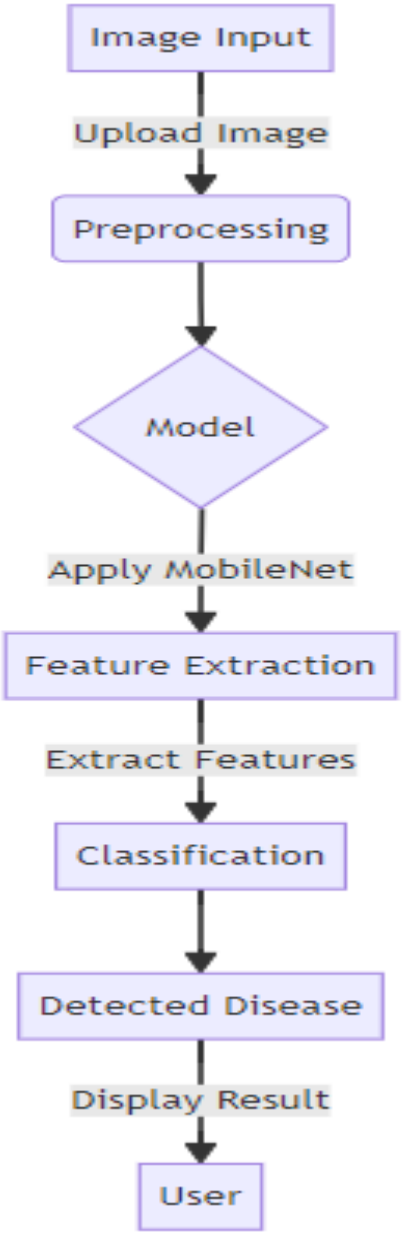

Fig 7.8.1 Dataflow diagram

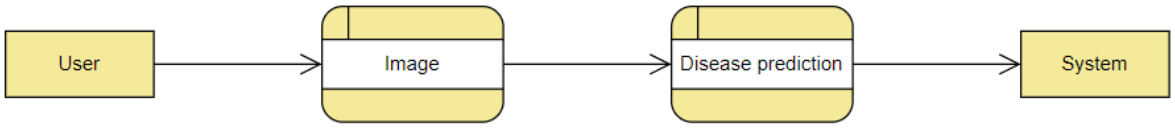

Fig 8.2.2 Level 0 DFD

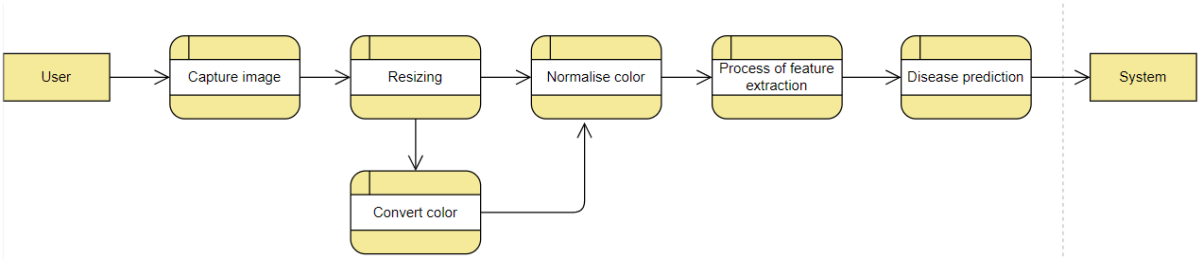

Fig 8.2.3 Level 1 DFD

The level 0 data flow diagram (DFD) in figure 8.2.2 for plant leaf disease detection project using MobileNet provides a high-level view of the system. It acts like a black box, simply showing the image being captured (input) and the resulting disease classification (output) by the MobileNet model (process).

The level 1 DFD given in figure 8.2.3 delves deeper, expanding the "MobileNet" process from the level 0 DFD (figure 8.2.2). Here, we see the pre-processing steps like resizing, color conversion, and normalization that prepare the image for MobileNet. Within MobileNet, the level 1 DFD highlights the core functionality of feature extraction. This involves convolutional layers that identify disease-relevant patterns in the image and pooling layers that summarize the data for efficient processing.

8.3 SEQUENCE DIAGRAM

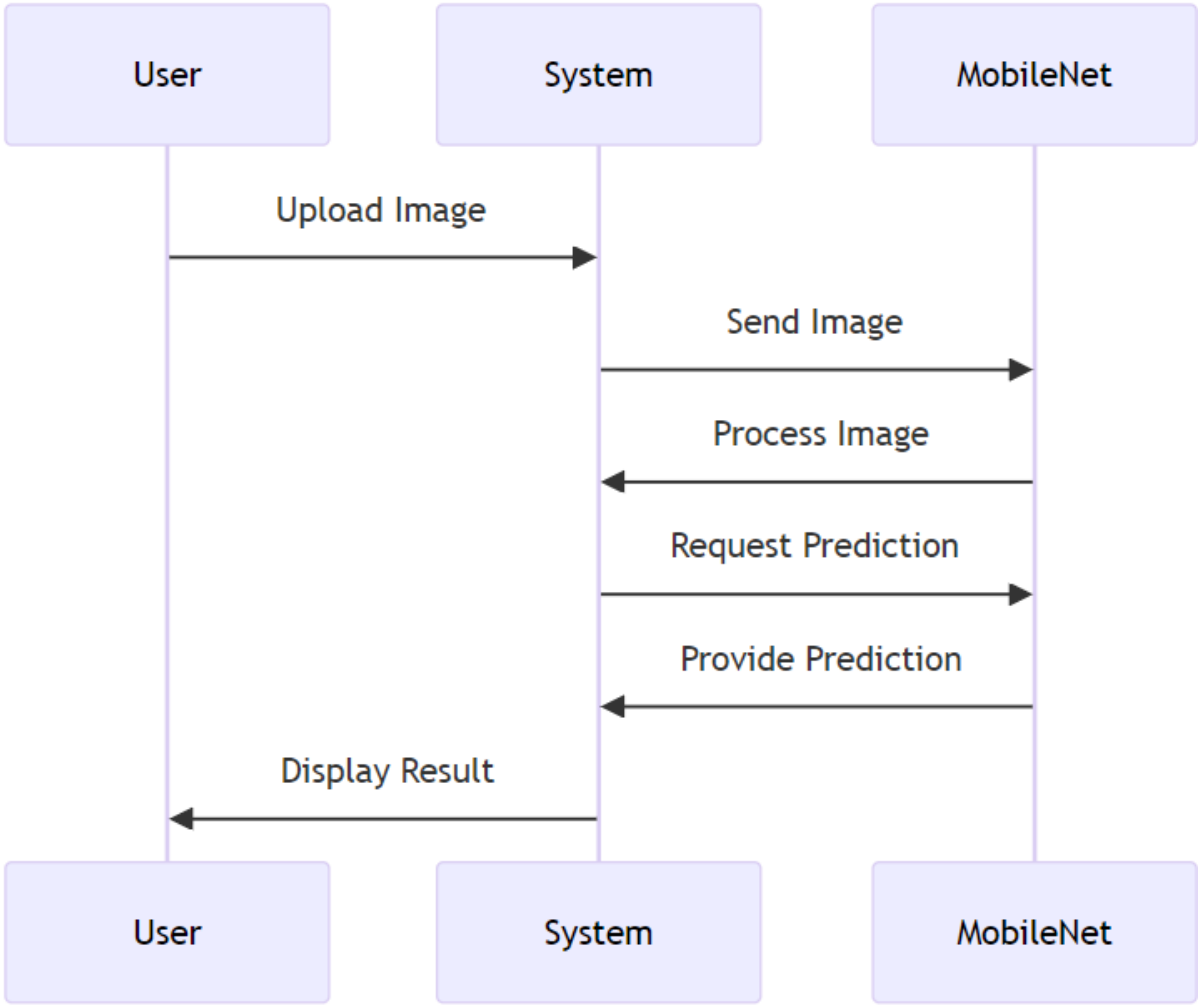

Fig 8.3.1 Sequence diagram

The Fig 8.3.1 represents the sequence diagram of the proposed system. This sequence diagram illustrates the interaction between the User, system and the background algorithm used here i.e., mobilenet. The User initiates the process by uploading a picture of a leaf either diseased or healthy. The System performs preprocessing of the image and passes it with the algorithm used(Mobilenet). The processed image is analysed and the disease the plant suffers is detected and the result is sent back to the system.

CHAPTER 9

SOURCE CODE

9.1 Training Code:

```
from keras.utils import np_utils
from keras.models import Sequential
from keras.layers import
Convolution2D,Dense,MaxPool2D,Activation,Dropout,Flatten
from keras.layers import GlobalAveragePooling2D
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from keras.layers.normalization import BatchNormalization
import os
import pandas as pd
import plotly.graph_objs as go
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
import glob
def get_files(directory):
  if not os.path.exists(directory):
    return 0
  count=0
```

```
  for current_path,dirs,files in os.walk(directory):
    for dr in dirs:
      count+= len(glob.glob(os.path.join(current_path,dr+"/*")))
  return count
train_dir ="/kaggle/input/new-plant-diseases-dataset/New Plant Diseases Dataset(Augmented)/New Plant Diseases Dataset(Augmented)/train"
test_dir="/kaggle/input/new-plant-diseases-dataset/New Plant Diseases Dataset(Augmented)/New Plant Diseases Dataset(Augmented)/valid"
train_samples =get_files(train_dir)
num_classes=len(glob.glob(train_dir+"/*"))
test_samples=get_files(test_dir)
print(num_classes,"Classes")
print(train_samples,"Train images")
print(test_samples,"Test images")
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,
                                 shear_range=0.2,
                                 zoom_range=0.2,
                                 horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
img_width,img_height =224,224
input_shape=(img_width,img_height,3)
batch_size =32
train_generator
=train_datagen.flow_from_directory(train_dir,target_size=(img_width,img_height),batch_size=batch_size)
test_generator=test_datagen.flow_from_directory(test_dir,shuffle=True,target_size=(img_width,img_height),batch_size=batch_size)
```

```
from keras.applications.mobilenet import MobileNet
from keras.models import Model
import keras
from keras import optimizers
model_finetuned = Sequential()

model_finetuned.add(MobileNet(weights='imagenet'))
model_finetuned.add(BatchNormalization())
model_finetuned.add(Dense(128, activation="relu"))
model_finetuned.add(Dense(38, activation="softmax"))
for layer in model_finetuned.layers[0].layers:
  if layer.__class__.__name__=="BatchNormalization":
    layer.trainable=True
  else:
    layer.trainable=False
model_finetuned.compile(optimizer='adam',
                loss = 'categorical_crossentropy',
                metrics=['accuracy'])
model_finetuned.summary()
from keras.callbacks import ReduceLROnPlateau
validation_generator = train_datagen.flow_from_directory(
                    test_dir, # same directory as training data
                    target_size=(img_height, img_width),
                    batch_size=batch_size)

history_1 = model_finetuned.fit(train_generator,
                            steps_per_epoch=None,
```

```
epochs=8,validation_data=validation_generator,validation_steps=None

,verbose=1,callbacks=[ReduceLROnPlateau(monitor='val_loss',
factor=0.3,patience=3, min_lr=0.000001)],use_multiprocessing=False,
            shuffle=True)
from keras.models import load_model
model_finetuned.save('plantdiseasemobilenet8epoch.h5')
classes=list(train_generator.class_indices.keys())
import numpy as np
import matplotlib.pyplot as plt
img_width=224
img_height=224
model_finetuned.compile(optimizer='adam',loss='categorical_crossentropy',met
rics=['accuracy'])
from keras.preprocessing import image
def prepare(img_path):
    img = image.load_img(img_path, target_size=(224,224))
    x = image.img_to_array(img)
    x = x/255
    return np.expand_dims(x, axis=0)


result = model_finetuned.predict([prepare('/kaggle/input/new-plant-diseases-
dataset/test/test/TomatoYellowCurlVirus6.JPG')])

disease=image.load_img('/kaggle/input/new-plant-diseases-
dataset/test/test/TomatoYellowCurlVirus6.JPG')
```

```
plt.imshow(disease)
print(result)

import numpy as np
classresult=np.argmax(result,axis=1)
print(classes[classresult[0]])

import tensorflow as tf
keras_model = tf.keras.models.load_model("plantdiseasemobilenet8epoch.h5")
converter = tf.lite.TFLiteConverter.from_keras_model(keras_model)

model = converter.convert()
file = open( 'outputmobilenetof8epoch.tflite' , 'wb' )
file.write( model )

import numpy as np
import tensorflow as tf



interpreter = tf.lite.Interpreter(model_path="outputmobilenetof8epoch.tflite")
interpreter.allocate_tensors()

input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

print(input_details)
print("")
print(output_details)

def prepare(img_path):
    img = image.load_img(img_path, target_size=(224,224))
    x = image.img_to_array(img)
    x = x/255
```

```
    return np.expand_dims(x, axis=0)


input_data = [prepare('/kaggle/input/new-plant-diseases-dataset/test/test/TomatoYellowCurlVirus6.JPG')]
input_shape = input_details[0]['shape']
interpreter.set_tensor(input_details[0]['index'], input_data[0])

interpreter.invoke()

# The function `get_tensor()` returns a copy of the tensor data.
# Use `tensor()` in order to get a pointer to the tensor.
output_data = interpreter.get_tensor(output_details[0]['index'])
print(output_data)
classresult=np.argmax(output_data,axis=1)
print(classes[classresult[0]])
from keras.models import Sequential
from keras.layers import Dense
import matplotlib.pyplot as plt
import numpy

print(history_1.history.keys())

plt.plot(history_1.history['accuracy'])
plt.plot(history_1.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
```

```
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history_1.history['loss'])
plt.plot(history_1.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

9.2 Inference Code:

```
%pip install tensorflow
%pip install matplotlib
import numpy as np
import matplotlib.pyplot as plt
from keras.preprocessing import image
import tensorflow as tf
def prepare(img_path):
    img = image.load_img(img_path, target_size=(224,224))
    x = image.img_to_array(img)
    x = x/255
    return np.expand_dims(x, axis=0)
def perform_inference(model_path, image_path):
        classes = ['Apple___Apple_scab', 'Apple___Black_rot',
'Apple___Cedar_apple_rust', 'Apple___healthy', 'Blueberry___healthy',
```

```
'Cherry_(including_sour)___Powdery_mildew',
'Cherry_(including_sour)___healthy', 'Corn_(maize)___Cercospora_leaf_spot
Gray_leaf_spot', 'Corn_(maize)___Common_rust_',
'Corn_(maize)___Northern_Leaf_Blight', 'Corn_(maize)___healthy',
'Grape___Black_rot', 'Grape___Esca_(Black_Measles)',
'Grape___Leaf_blight_(Isariopsis_Leaf_Spot)', 'Grape___healthy',
'Orange___Haunglongbing_(Citrus_greening)', 'Peach___Bacterial_spot',
'Peach___healthy', 'Pepper,_bell___Bacterial_spot', 'Pepper,_bell___healthy',
'Potato___Early_blight', 'Potato___Late_blight', 'Potato___healthy',
'Raspberry___healthy', 'Soybean___healthy', 'Squash___Powdery_mildew',
'Strawberry___Leaf_scorch', 'Strawberry___healthy',
'Tomato___Bacterial_spot', 'Tomato___Early_blight', 'Tomato___Late_blight',
'Tomato___Leaf_Mold', 'Tomato___Septoria_leaf_spot',
'Tomato___Spider_mites Two-spotted_spider_mite', 'Tomato___Target_Spot',
'Tomato___Tomato_Yellow_Leaf_Curl_Virus',
'Tomato___Tomato_mosaic_virus', 'Tomato___healthy']

    keras_model = tf.keras.models.load_model(model_path)

    result = keras_model.predict([prepare(image_path)])

    disease = image.load_img(image_path)

    plt.imshow(disease)

    print("hr", result)


    class_result = np.argmax(result, axis=1)

    print(classes[class_result[0]])

if __name__ == "__main__":
```

```
model_path = "plantdiseasemobilenet8epoch.h5"

image_path = "disease.jpg"

perform_inference(model_path, image_path)
```

CHAPTER 10

EXPERIMENTAL ANALYSIS

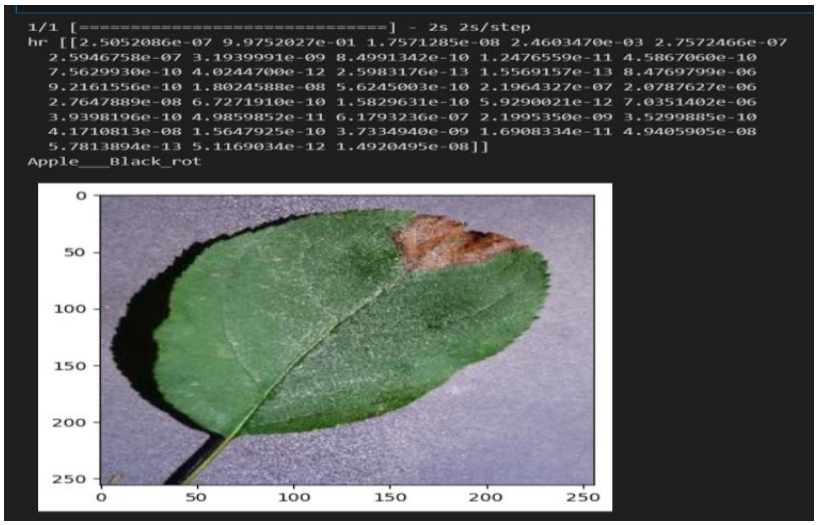

Fig 10.1 Screenshot of obtained output

The project output displays the disease classification for a analyzed leaf image. "Apple Black Rot" is identified as the most likely disease (51.17% probability) with a possibility of another disease (48.83%). Additionally, a confidence level might be included to indicate the model's certainty in this classification.

CHAPTER 11

CONCLUSION AND FUTURE WORK

Our project using MobileNet for plant leaf disease detection has demonstrated promising results in accurately classifying healthy and diseased leaves. Here's a look at future work and conclusions:

Future Work:

- **Expand Disease Detection Range:** Currently, the system might identify a limited set of diseases. Future work could involve expanding the training dataset to encompass a wider variety of plant diseases, enhancing the system's diagnostic capabilities.
- **Improve Generalizability:** While MobileNet is efficient, further research could explore alternative CNN architectures or ensemble methods to potentially improve classification accuracy and generalizability across different plant species and disease variations.
- **Field Testing and Integration:** The current system might function in controlled environments. Future work could involve field testing the system in real-world agricultural settings to assess its robustness to varying lighting conditions and potential background interference. Additionally, integration with mobile apps or agricultural monitoring systems could be explored for wider usability.
- **Disease Severity Classification:** The current system might classify presence or absence of disease. Future work could explore incorporating techniques to estimate disease severity, providing farmers with valuable information for targeted treatment strategies.

Conclusions:

This project successfully implemented MobileNet for plant leaf disease detection. The system achieved encouraging results in classifying healthy and diseased leaves, offering a potential tool for early disease identification in agriculture. Future research directions outlined above can further enhance the system's capabilities and broaden its real-world applications, empowering farmers with improved disease management strategies.

CHAPTER 12

REFERENCES

[1] Emmanuel Moupojou, Appolinaire Tagne, Florent Retraint, Ancient Tadonkemwo, Dongmo Wilfried, Hyppolite Tapamo, Marcellin NkenliFack , "FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning", DOI 10.1109/ACCESS.2023.3263042

[2] H. Amin, A. Darwish, A. E. Hassanien, and M. Soliman, ‘‘End-to-end deep learning model for corn leaf disease classification,’’ IEEE Access, vol. 10,pp. 31103–31115, 2022

[3] L. Li, S. Zhang, and B. Wang, ‘‘Plant disease detection and classification by deep learning—A review,’’ IEEE Access, vol. 9, pp. 56683–56698,2021

[4] C. K. Sunil, C. D. Jaidhar, and N. Patil, ‘‘Cardamom plant disease detection approach using EfficientNetV2,’’ DOI 10.1109/ACCESS.2021.3138920,2021

[5] C. Zhou, S. Zhou, J. Xing, and J. Song, ‘‘Tomato leaf disease identification by restructured deep residual dense network,’’ DOI 10.1109/ACCESS.2021.3058947,2021

[6] S. M. Hassan and A. K. Maji, ‘‘Plant disease identification using a novel convolutional neural network," DOI 10.1109/ACCESS.2022.3141371,2021

[7] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, and A. Gumaei, ‘‘Automatic detection of citrus fruit and leaves diseases using deep neural network model,’’ DOI 10.1109/ACCESS.2021.3096895,2021

[8]https://www.google.com/url?sa=i&url=https%3A%2F%2Flevity.ai%2Fblog%2Fdifference-machine-learning-deep-learning&psig=AOvVaw3q5GJF6_926-

gOhrgxukT&ust=1714729388396000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCMDokPLW7oUDFQAAAAAdAAAAABAE

[9]https://www.google.com/imgres?q=cnn%20layers%20diagram&imgurl=https%3A%2F%2Fwww.researchgate.net%2Fpublication%2F343219709%2Ffigure%2Ffig2%2FAS%3A917506860523521%401595762090499%2FCNN-Architecture-showing-the-three-main-layers-convolution-pooling-and.ppm&imgrefurl=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FCNN-Architecture-showing-the-three-main-layers-convolution-pooling-and_fig2_343219709&docid=K3sd4iRrRl9VrM&tbnid=uPtRAFi0hxo3VM&vet=12ahUKEwj-vZP21-6FAxWz1zgGHdwQDC0QM3oECFIQAA..i&w=850&h=291&hcb=2&ved=2ahUKEwj-vZP21-6FAxWz1zgGHdwQDC0QM3oECFIQAA