A COMPREHENSIVE STUDY OF LEAF DISEASE IDENTIFICATION ON TOMATO AND POTATO PLANTS

A PROJECT REPORT

Submitted by

AKSH KALYANI [RA2011003010525]
NIPUN CHAURASIA [RA2011003010536]

Under the Guidance of

Dr. S. PRIYA

(Assistant Professor (Sr.G), Department of Computing Technologies)

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

DEPARTMENT OF COMPUTING TECHNOLOGIES

COLLEGE OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR- 603203

NOVEMBER 2023

Department of Computing Technologies

SRM Institute of Science & Technology

Own Work Declaration Form

Degree/ Course : B.Tech in Computer Science and Engineering

Student Name : AKSH KALYANI, NIPUN CHAURASIA

Registration Number : RA2011003010525, RA2011003010536

Title of Work : A COMPREHENSIVE STUDY OF LEAF DISEASE IDENTIFICATION ON TOMATO AND POTATO PLANTS.

I / We hereby certify that this assessment compiles with the University's Rules and Regulations relating to Academic misconduct and plagiarism**, as listed in the University Website, Regulations, and the Education Committee guidelines.

I / We confirm that all the work contained in this assessment is my / our own except where indicated, and that I / We have met the following conditions:

- Clearly referenced / listed all sources as appropriate
- Referenced and put in inverted commas all quoted text (from books, web, etc)
- Given the sources of all pictures, data etc. that are not my own
- Not made any use of the report(s) or essay(s) of any other student(s) either past or present
- Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources)
- Compiled with any other plagiarism criteria specified in the Course handbook / University website

I understand that any false claim for this work will be penalized in accordance with the University policies and regulations.

DECLARATION:

I am aware of and understand the University's policy on Academic misconduct and plagiarism and I certify that this assessment is my / our own work, except where indicated by referring, and that I have followed the good academic practices noted above.

AKSH KALYANI SIGNATURE:

NIPUN CHAUASIA SIGNATURE:

DATE: 08-11-2023

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY
KATTANKULATHUR–603 203

BONAFIDE CERTIFICATE

Certified that 18CSP109L / I8CSP111L project report titled "**A Comprehensive Study on Leaf Disease Detection on Potato and Tomato Plants**" is the bonafide work of **AKSH DHARMESH KALYANI [Reg No. RA2011003010525] and NIPUN CHAURASIA [Reg No. RA2011003010536]** who carried out the project work under my supervision. Certified further, that to the best of my knowledge, the work reported here in does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr. S. PRIYA
SUPERVISOR
Assistant Professor
Department of Computing Technologies

Dr. T. MANORANJITHAM
PANEL HEAD
Associate Professor
Department of Computing Technologies

Dr. M. PUSHPALATHA
HEAD OF THE DEPARTMENT
Department of Computing Technologies

ACKNOWLEDGEMENT

We express our humble gratitude to **Dr. C. Muthamizhchelvan**, Vice-Chancellor, SRM Institute of Science and Technology, for the facilities extended for the project work and his continued support.

We extend our sincere thanks to Dean-CET, SRM Institute of Science and Technology, **Dr. T. V. Gopal**, for his invaluable support.

We wish to thank **Dr. Revathi Venkataraman**, Professor and Chairperson, School of Computing, SRM Institute of Science and Technology, for her support throughout the project work.

We are incredibly grateful to our Head of the Department, **Dr. M. Pushpalatha**, Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for her suggestions and encouragement at all the stages of the project work.

We want to convey our thanks to our Project Coordinators, Panel Head, **Dr. T. Manoranjitham**, Associate Professor and Panel Members, **Dr. J. Kalaivani**, Assistant Professor, **Dr. S. Vidhya** Assistant Professor and **Dr. B. Prakash** Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for their inputs during the project reviews and support.

We register our immeasurable thanks to our Faculty Advisor, **Mrs. R. Brindha**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for leading and helping us to complete our course.

Our inexpressible respect and thanks to our guide, **Dr. S. Priya**, Assistant Professor, Department of Computing Technologies, SRM Institute of Science and Technology, for providing us with an opportunity to pursue our project under her mentorship. She provided us with the freedom and support to explore the research topics of our interest. Her passion for solving problems and making a difference in the world has always been inspiring.

We sincerely thank all the staff and students of Computing Technologies Department, School of Computing, S.R.M Institute of Science and Technology, for their help during our project. Finally, we would like to thank our parents, family members, and friends for their unconditional love, constant support and encouragement.

AKSH KALYANI [Reg. No: RA2011003010525]

NIPUN CHAURASIA [Reg. No: RA1911003010536]

ABSTRACT

In India, where almost 70% of the population depends on agriculture for a living, In order to reduce the crop losses, plant diseases must be identified as soon as possible. On the other hand, manual disease monitoring is time-consuming, labor-intensive, and demands specific knowledge. This research work uses machine learning and image processing methods to identify plant illnesses from leaf photos in an effort to overcome these difficulties. Leaf images are processed using image processing, a subset of signal processing, to extract pertinent information. Automatic decision-making based on training data is made possible by machine learning, a branch of artificial intelligence. The main goal of research is to classify diseases using characteristics including leaf color, damage level, leaf area, and textural parameters. Computer vision and machine learning techniques to propose a practical and effective method for disease identification in agriculture. The method addresses the labor-intensive and time-consuming aspect of disease surveillance in agriculture with an amazing 94% accuracy in detecting 13 different illnesses across potato and tomato plant species.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Table No. **Table Name** **Page No.**

ABBREVIATIONS

AI	Artificial Intelligence
DNN	Deep Neural Network
DL	Deep Learning
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
GLCM	Gray Level Cooccurrence Matrix
CGAN	Conditional Generative Adversarial Network
DCGAN	Deep Convolutional Generative Adversarial Network
ANN	Artificial Neural Networks

CHAPTER 1

INTRODUCTION

1.1. INTRODUCTION

India heavily relies on agriculture, with approximately 70% of its population engaged in this vital sector. However, the identification of plant diseases poses a significant challenge, as manual inspection demands an enormous workforce with expertise in plant diseases and consumes substantial time. To address this challenge, we turn to the synergy of image processing and machine learning models for plant disease detection. In this project, we present a method for detecting plant diseases through the analysis of leaf images.

Image processing, a specialized field within signal processing, empowers us to extract valuable image properties and information. Complementing this, machine learning, a subset of artificial intelligence, offers automated learning and instruction capabilities to perform specific tasks. The primary objective of machine learning is to comprehend training data and create models that are beneficial for decision-making and the accurate prediction of outcomes using extensive training data.

Our approach centers on the analysis of various features derived from leaf images, including leaf color, extent of damage, leaf area, and textural attributes, for the purpose of disease classification. This comprehensive analysis allows us to effectively identify diverse plant leaf diseases with the utmost accuracy.

Traditionally, plant disease detection relied on visual leaf inspection or involved complex chemical processes conducted by experts. This method necessitated large teams of experts and constant plant monitoring, which could be prohibitively expensive when applied to extensive farmlands. In this context, our proposed system offers a valuable solution for monitoring large plant fields. By automatically detecting diseases based on visible symptoms on plant leaves, our system simplifies and economizes the process. Furthermore, our approach stands out for its computational efficiency and reduced prediction time compared to other deep learning-based methods, as it leverages statistical machine learning and image processing algorithms.

1.2. PROBLEM STATEMENT

In India, there is a wide variety of plants cultivated by farmers, and the traditional method of detecting and recognizing plant diseases relies on visual observation with the naked eye. This approach is a gradual process and often lacks certainty. Particularly in rural areas of India, consulting experts to identify plant diseases can be expensive and time-consuming due to the limited availability of such experts.

Managing plant diseases in large farms requires multiple teams of experts and ongoing monitoring, incurring significant costs. Excessive use of pesticides can also be detrimental to natural resources like water, soil, air, and the food chain. Minimizing pesticide contamination in food products is essential. Therefore, there is a need for automated systems to detect plant diseases early on.

The most common method for identifying plant diseases is by examining the leaves. Common diseases include brown and yellow spots, early and late scorch, as well as fungal, viral, and bacterial diseases. Automated detection methods are essential for early symptom recognition and efficient disease management.

1.3. OBJECTIVE

The objective of plant disease prediction utilizing Artificial Intelligence and deep neural networks is to provide a more precise and effective method of identifying agricultural illnesses, allowing farmers to take preventative actions to safeguard their plants and improve production in terms of both quantity and quality. Traditional identification of illness techniques, such as visual inspection, can be labor- and time-intensive and often not particularly accurate. Data may be gathered and analyzed in real-time by combining Artificial Intelligence technology and deep neural networks, giving farmers important knowledge on the health of their plants.

The initial goal is to create an Artificial Intelligence-based sensor network that can be used to gather information on various plant health-related metrics, including temperature, humidity, and soil moisture levels. Deep neural networks can then be used to analyze this data and find patterns and trends in plant diseases. Deep neural networks can precisely detect and forecast agricultural illnesses using image processing and machine learning methods, enabling farmers to take preventive action to save their plants.

The objectives of Plant Disease Prediction Using Artificial Intelligence and Deep Neural Networks are as follows:

1. **Create an Artificial Intelligence-based sensor network to gather data:** The initial goal is to establish a sensor network that can collect data on various aspects of plant health, such as temperature, humidity, and soil moisture levels. This information is crucial for monitoring the condition of plants and identifying any signs of disease. The sensor network consists of sensors distributed across the farm, continuously collecting data. This data is then sent to a central hub using Artificial Intelligence technology, enabling real-time monitoring of plant health. This real-time monitoring allows for the early detection of disease symptoms and enables farmers to take prompt action to protect their plants. Additionally, the data collected can be used to identify trends and patterns in plant diseases, helping farmers develop more effective disease management plans.

2. **Analyze data using deep neural networks:** The second goal is to leverage deep neural networks to analyze the data collected from the sensor network. Deep neural networks are capable of accurately detecting and predicting agricultural diseases using image processing and machine learning techniques. The data collected, which is typically in the form of images or time-series data (e.g., temperature and humidity readings), is processed by deep neural networks. These networks consist of layers of artificial neurons that learn patterns and relationships between different variables. Once trained on a dataset of images or time-series data, deep neural networks can be used to predict the presence of plant diseases in new data.

3. **Develop a user-friendly interface**: The third goal is to provide a user-friendly interface that farmers can use to access real-time information about their plants. This interface should offer comprehensive information about plant health, including disease symptoms, and provide recommendations on preventive measures, such as adjusting watering schedules or using fungicides. Developing a user-friendly interface is crucial for the

success of plant disease prediction using Artificial Intelligence and deep neural networks. It allows farmers to quickly and efficiently access essential information about their plant health, enabling timely action to prevent disease spread and plant damage. A user-friendly interface also encourages the adoption of this technology among farmers, ultimately leading to increased plant yields and improved plant quality.

4. **Improve the efficiency and sustainability of farming operations:** The ultimate goal is to provide farmers with fast and accurate information to enhance the sustainability and efficiency of farming operations. Plant diseases can significantly impact the productivity and sustainability of farming operations, leading to yield and quality losses, increased pesticide and fertilizer usage, and reduced profitability. By utilizing Artificial Intelligence and deep neural networks, farmers can preserve their plants, increase operational efficiency and sustainability, and accurately predict plant diseases.

   This approach reduces the time and resources required for disease identification and control, thus increasing efficiency. Traditional disease detection methods, relying on visual inspection, are labor- and time-intensive. However, Artificial Intelligence-based sensor networks and deep neural networks enable faster and more accurate disease identification, allowing farmers to take timely preventive action.

5. **Facilitate decision-making for farmers:** One of the major challenges farmers face is making decisions on how to manage their plants amid uncertainty. Plant diseases can be unpredictable and spread rapidly, making it challenging for farmers to determine when and how to take action. To address this challenge, deep neural networks and Artificial Intelligence are used to facilitate decision-making for farmers. The system should provide farmers with real-time information about their plant health and the presence of any diseases.

   This information must be highly specific, accurate, and presented in a simple and understandable manner. Visualizations and alerts can draw attention to potential problems. With access to this information, farmers can make informed decisions about how to manage their plants, whether it involves using fungicides, altering irrigation methods, or adjusting planting schedules to prevent disease spread. In summary, the objective of using Artificial Intelligence and deep neural networks in plant disease prediction is to provide a more accurate and efficient method for identifying and managing agricultural illnesses. This approach supports early disease detection, enhances operational efficiency and sustainability, and facilitates informed decision-making for farmers. By leveraging the power of Artificial Intelligence, farmers can protect their plants, optimize plant production, and contribute to sustainable and productive agriculture.

1.4. SCOPE

The scope of "plant disease prediction using Artificial Intelligence and deep neural networks" is to accurately forecast plant diseases by utilizing deep neural networks that have been trained using data from Artificial Intelligence devices and sensors to monitor plant health and environmental variables. Early disease diagnosis, enhanced plant output, reduced pesticide usage, and increased agricultural efficiency are all benefits of this strategy. It has great potential to advance sustainable farming methods and has a favorable influence on the agricultural sector. In the old days, farmers would personally check their plants for disease symptoms and then take appropriate action depending on their findings. It requires an extensive amount of work and attention to detail to use this strategy to identify illnesses at an early stage when they are most curable.

Real-time data collection on plant health and environmental variables is possible with Artificial Intelligence devices, sensors, and deep neural networks. Temperature, humidity, soil moisture, and other elements that may have an impact on plant development are among the data that sensors may collect. To identify possible illnesses and forecast their spread, this data may be put into deep neural networks that have been trained on massive databases of plant disease trends. This strategy has a lot of advantages. Farmers who can identify infections early can act rapidly to stop the spread of the illness and minimize plant losses.

This strategy may also result in more environmentally friendly agricultural methods by lowering the usage of pesticides and other chemicals, which may be expensive for farmers and have negative environmental effects. By making educated choices about irrigation, fertilization, and harvesting, farmers may optimize plant productivity and decrease waste with the aid of accurate disease prediction.

The following are some of the major spheres where this technology can make a difference:

1. **Early disease detection:** Using Artificial Intelligence and deep neural networks to forecast agricultural illnesses has several advantages, one of which is the capacity to identify diseases at an early stage. This can assist farmers in acting swiftly to stop the disease's spread and reduce plant losses. Deep neural networks are used to analyze data after sensors continually monitor plants in order to find patterns and anomalies that could point to the existence of illness. A major advantage of plant disease prediction using Artificial Intelligence and deep neural networks is early disease identification.

   Detecting diseases promptly enables landowners to act swiftly to halt illness's spread or reduce plant losses. This is crucial since it can be challenging and expensive to control a disease after it has established itself. Artificial Intelligence-based sensors may be installed in the field to continually monitor plant health while gathering information on the weather, soil moisture, and other important factors. Deep neural networks are then used to analyze this data in order to find patterns and anomalies that could point to their existence of illness.

   Deep neural networks have the benefit of processing enormous volumes of data and spotting tiny patterns that human observers would miss. This implies that even small changes in plant health can be recognized, acting as a possible disease early warning system.

2. **Precision agriculture:** Precision agriculture, which uses data-driven methods to optimize plant output, is a bigger trend that includes plant disease prediction utilizing Artificial Intelligence and deep neural networks. This technology can assist to increase plant yields and quality while lowering resource waste by giving farmers precise and timely information on the health of their plants. Precision agriculture is a method of farming that maximizes plant output, minimizes waste, and lessens the impact of farming on the environment. Because it gives farmers precise and timely information about plant health, plant disease prediction using Artificial Intelligence and deep neural networks is a crucial part of precise farming.

Information about weather, moisture, soil wetness, and other environmental elements that might have an impact on plant development and health may be collected using Artificial Intelligence-based sensors. Deep neural networks are used to analyze this data and find patterns and abnormalities that might be signs of illness. Farmers may maximize plant yield and minimize waste by utilizing this data to make educated decisions regarding irrigation, fertilization, and other inputs. Using Artificial Intelligence and deep neural networks to anticipate plant diseases can also assist farmers in minimizing the use of pesticides and other chemicals, which can be detrimental to the environment and public health.

Farmers may decrease the need for pesticides and other chemicals, resulting in more sustainable and ecologically friendly agricultural practices, by identifying illnesses early and taking the right measures. As a result, farming operations become more effective and productive by enabling farmers to make better decisions about planting, harvesting, and other tasks.

3. **Sustainable agriculture:** Plant disease prediction utilizing Artificial Intelligence and deep neural networks can help create a more sustainable agricultural system by increasing the effectiveness of farming operations and lowering the usage of chemical inputs. In light of climate change and mounting demands on natural resources, this is especially crucial. Another crucial use of Artificial Intelligence and deep neural networks for plant disease prediction is sustainable agriculture. Sustainable agriculture is the practice of growing plants while reducing their harmful effects on the environment and enhancing social and economic well-being.

   Artificial Intelligence and deep neural network-based plant disease prediction can support sustainable agriculture in a number of ways. Farmers can minimize plant losses and cut back on the use of pesticides and other chemicals via early and precise disease detection. This can lessen the detrimental effects on the health of agricultural workers and the local population while preserving the natural ecology and biodiversity of the surrounding area.

   Plant disease prediction utilizing Artificial Intelligence and deep neural networks can aid farmers in minimizing the usage of pesticides while simultaneously maximizing the utilization of water, fertilizer, and other inputs. Farmers may identify fields that need more or less water, fertilizer, or other inputs by using data collected by Artificial Intelligence sensors and analyzed by deep neural networks, minimizing waste and promoting more effective resource use.

In order to be sustainable, agriculture must also support social and economic well-being. Plant disease prediction utilizing Artificial Intelligence and deep neural networks can help farmers boost their yields and revenues, improving economic viability by minimizing plant losses and lowering waste.

4. **Improve the efficiency and sustainability of farming operations:** The ultimate goal is to give farmers fast and accurate information in order to increase the sustainability and efficiency of farming operations. We can increase the effectiveness and sustainability of farming operations while lowering costs and environmental impact by enabling farmers to make educated decisions. One of the main goals of plant disease prediction utilizing Artificial Intelligence and deep neural networks is to boost performance and sustainability of agricultural operations.

   Utilizing these technologies, farmers can choose their plants with more knowledge, use resources more effectively, and produce less waste, leading to more productive and environmentally friendly agricultural methods. Reducing the need for human labor is one way plant disease prediction may boost farming operations' efficiency. Traditional disease detection methods rely on visual inspection by skilled professionals, which may be labor- and time-intensive.

   Farmers may lessen the need for physical inspection by employing Artificial Intelligence sensors and deep neural networks to identify illnesses early and correctly, and instead concentrate their efforts on adopting preventative actions to safeguard their plants. Predicting plant diseases can also assist farmers in making the most use of resources like water, fertilizer, and pesticides. Farmers may identify fields that need more or less water, fertilizer, or pesticides by analyzing data from Artificial Intelligence sensors and deep neural networks, minimizing waste and promoting more effective resource usage. This can not only raise the profitability and sustainability of farming operations but also cut expenditures.

1.5. INNOVATION

The novelty in the project "Plant Disease Prediction Using A.I and Deep Neural Networks" lies in the application of A.I and deep learning models for precise and effective identification of plant diseases.

The development of this system aims to prevent plant losses due to diseases and offers a practical solution to assist farmers. A notable advancement is the utilization of the Reference Architecture for Artificial Intelligence, enabling cloud-based and on-field disease detection algorithms for disease anticipation. Another innovative aspect enhancing prediction accuracy is the use of neural networks for training the system on disease categorization and prevention.

Moreover, the standardization of data and the extraction of pixel characteristics from photos taken by farmers, narrowing down the range of output values, represent a creative approach that improves the system's effectiveness. Another novel feature is the neural network classification assisted by back-propagation, providing farmers with diagnosis and treatment recommendations. To enhance user-friendliness, the system incorporates farmer forums for localized discussions and supports local languages, as well as a speech-to-text capability in the app.

The system has been trained on disease categorization and prevention using neural networks, consistently achieving high accuracy rates of 93% or more across various plant models, a crucial aspect for machine learning algorithms. Early disease detection, enabling timely protective measures for plants, is a key benefit of this innovation. Consequently, it leads to increased plant yields and higher-quality produce, contributing to improved food security. By reducing the time and labor required compared to traditional disease diagnosis methods, this technology also has the potential to boost the efficiency and sustainability of agricultural operations.

Plant Disease Prediction Using A.I and Deep Neural Networks innovatively combines deep neural networks with A.I technology to accurately and effectively identify plant diseases. By facilitating early disease diagnosis, improving the sustainability and efficiency of farming operations, and prioritizing accessibility and inclusivity for farmers, the system has the potential to bring about a profound transformation in the agricultural sector.

CHAPTER 2

LITERATURE SURVEY

2.1. OVERVIEW

Plant diseases happen to be a major threat to agricultural efficiency and productivity, with the potential for substantial economic losses and decreased plant yields. Traditional methods of the prevention and detection of plant disease usually rely on visual inspection, a process that is both time-consuming and prone to inaccuracies. In response to these challenges, emerging technologies, such as deep neural networks and the Artificial Intelligence (A.I), offer novel opportunities for more precise and efficient plant disease detection, with a specific focus on the leaves of potato and tomato plants.

Traditional disease diagnosis methods for these plants, like expert visual assessment, can be resource-intensive, error-prone, and may not always yield timely results. As a result, there is a growing need to explore innovative solutions that can enhance the accuracy and speed of disease identification in potato and tomato plants.

The role of deep neural networks in plant disease detection is paramount. These networks are inspired by the structure and function of the human brain and are well-suited for the analysis of large datasets, making them a promising tool for identifying diseases in plants. The use of A.I technology complements this approach, as it allows for the collection of real-time data from various sources, including environmental sensors, cameras, and other devices placed in agricultural fields. This data can offer crucial insights into the health of potato and tomato plants and help detect diseases promptly.

The integration of A.I devices and deep neural networks enables a more accurate and efficient disease detection process, reducing the reliance on manual examination and human judgment. By analyzing the collected data, these technologies can swiftly identify disease symptoms, variations in leaf conditions, and other vital indicators of plant health.

The significance of this project extends beyond agriculture. Accurate and timely disease detection on potato and tomato leaves can lead to informed decisions regarding the management of plants, the application of treatments, and ultimately, higher agricultural yields. Additionally, it can contribute to a reduction in the use of pesticides and resources, making agricultural practices more sustainable and environmentally friendly. However, there are challenges that need to be addressed. High-quality data collection, data privacy, and security issues, as well as the standardization of data formats, are important considerations. The success of this technology in plant disease detection hinges on overcoming these challenges and continued research efforts.

2.2. DEEP NEURAL NETWORKS IN PLANT DISEASE

Leaf disease detection has been revolutionized by deep neural networks (DNNs), which provide automated and accurate identification of plant diseases. DNNs excel at learning features, extracting relevant characteristics from leaf images, and adapting to the specific attributes of different plant species and diseases. Their scalability and ability to handle large datasets enable real-time disease monitoring when combined with cameras and A.I devices. Importantly, DNNs reduce the reliance on human expertise, making disease detection accessible for farmers and enabling early interventions.

DNNs are well-suited for dynamic agricultural environments due to their continual learning and adaptability. However, challenges such as the need for substantial datasets, infrastructure, and data security must be addressed. With ongoing research and development, DNNs have the potential to revolutionize plant health management, minimize losses, and promote sustainable agricultural practices.

Deep neural networks (DNNs) have emerged as a revolutionary technology in the realm of leaf disease detection, promising an automated and highly accurate approach to identifying plant diseases. The core principle behind the operation of DNNs in this context is their capacity to learn and recognize intricate patterns and features in leaf images, which may indicate the presence of diseases.

DNNs operate by leveraging multiple layers of artificial neurons that are structured to mimic the human brain's neural network. These artificial neurons analyze and process the features extracted from the images. During the training phase, DNNs are fed a substantial dataset of labeled images of healthy and diseased leaves. Through a process known as backpropagation, the network adjusts the weights of connections between neurons to minimize the difference between its predictions and the actual labels in the training data. This iterative learning process fine-tunes the network, allowing it to capture complex patterns and variations indicative of different diseases and leaf conditions.

One key advantage of DNNs is their ability to automatically extract relevant features from leaf images, including color variations, textures, and the spatial distribution of lesions or discolorations. This feature learning is crucial as it allows DNNs to adapt to the unique characteristics of various plant species and diseases. Moreover, they are highly scalable, capable of processing large datasets efficiently.

When integrated with hardware such as cameras and A.I devices, DNNs enable real-time disease monitoring. This means that as images of plant leaves are captured, the DNN can promptly analyze them and make disease identifications. Real-time monitoring is vital for early disease detection, which can significantly reduce the spread of diseases and minimize plant losses.

Furthermore, DNNs reduce the reliance on human expertise, which is often necessary in traditional disease diagnosis methods. This democratization of disease detection empowers farmers and agricultural stakeholders by providing them with a powerful tool to protect their plants. DNNs continue to improve over time as they encounter more data, which is essential in the dynamic agricultural landscape where disease patterns and strains may change.

While DNNs hold immense promise for leaf disease detection, they also present challenges. Large, high-quality datasets are necessary for training, and there are infrastructure requirements for image capture and analysis. Additionally, issues related to privacy and data security need to be addressed, particularly if cloud-based solutions are employed. As research and development in this field progresses, DNNs are poised to transform plant health management, minimize disease-related losses, and promote more sustainable and efficient agricultural practices.

2.3. COMPARISON OF EXISTING SYSTEM AND PROPOSED SYSTEM

Author	**S. Khirade et. Al. (2015)**	**Shiroop Madiwalar et Al. (2017)**	**Peyman Moghadam et Al. (2017)**	**Sharath D. M. et Al. (2019)**	**Garima Shrestha et Al. (2020)**	**Method Proposed**
Algorithm used	Digital image processing with BPNN	Digital Image processing with SVM	Hyperspectral imaging with SVM	Digita.1 Image processing	Convolutional Neural Networks	Digital image processing and random forest classifier
Accuracy	-	83.34%	93%	-	88.80%	93%
Computationally efficient	❌	✔	❌	✔	❌	✔
Specialized hardware requirement	❎	❎	✔	❎	❎	❎

Table 2.1 Comparison of existing and proposed system

2.3.1. EXISTING SYSTEM

The commonly employed techniques include the physical expansion approach, which involves adjustments such as tensile rotation, resolution picture translation, and disturbance. Additionally, the variational auto-encoder (VAE), web crawler, and autoregressive model are also frequently utilized methods. The classic expansion method is associated with several limitations, including subpar quality, insufficient diversity, and unevenness in the produced samples.

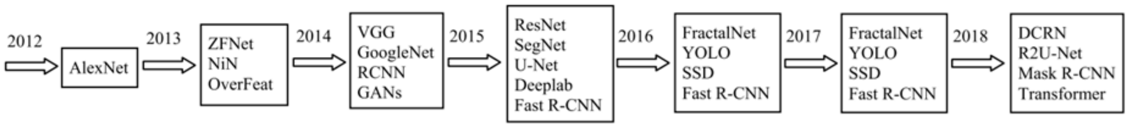


Fig 2.1. Traditional methods of image detection

2.3.2. PROPOSED SYSTEM GANs

Generative Adversarial Networks (GANs) represent a type of generative model that was introduced by Goodfellow et al. in 2014 [29]. Following then, a series of GAN versions have been introduced, including but not limited to DCGAN, LAPGAN, PGGAN, CGAN, F-GAN, WGAN, InfoGAN, SeqGAN, and LeakGAN. The primary objective is to produce artificial samples that possess identical features to the provided training distribution. The GAN models primarily comprise two components, namely the generator and the discriminator.

The structure diagram is depicted in Figure 2.2. Generative network methodologies have been widely employed for sample generation in recent times. The study conducted by Nazki et al. [30] is the pioneering effort in utilizing Generative Adversarial Networks (GANs) for the purpose of artificially enhancing the dataset in order to enhance the performance of plant disease recognition. By enhancing the activation reconstruction loss (ARL) function and proposing an enhanced AR-GAN model, the current study incorporates this model into composite images of nine different types of tomatoes from a test dataset consisting of 2789 samples. The findings indicate a notable increase in classification accuracy (+5.2%) compared to conventional approaches, thereby surpassing the performance of prominent existing models. Tian et al. introduced a methodology known as CycleGAN, which enables the generation of a greater number of images depicting apple diseases. The utilization of conditional deep convolutional generative adversarial networks (C-DCGAN) involves the augmentation of generated images, wherein the segmented tea illness spot image is employed as the input for VGG16. The findings indicate that the utilization of C-DCGAN yields an average accuracy that is approximately 28% greater compared to the application of rotation and translation techniques.

The study employed deep convolutional generative adversarial networks (DC-GAN) to produce visual representations, resulting in a notable top-1 average identification accuracy of 94.33% when evaluated on GoogLeNet. The T-distribution stochastic neighborhood embedding (T-SNE) technique demonstrated that the image distribution produced by this approach exhibited a higher degree of similarity to the distribution of the actual picture samples. The study conducted by the authors [34] involved the utilization of a novel Leaf GAN model to analyze and classify four distinct types of leaf diseases observed in grape plant photos. The empirical findings indicate that the Leaf GAN model effectively enhances the visibility of grape leaf diseases in photos and successfully generates a sufficient number of grape leaf disease images. The superiority of Leaf GAN over DCGAN and WGAN has been demonstrated.

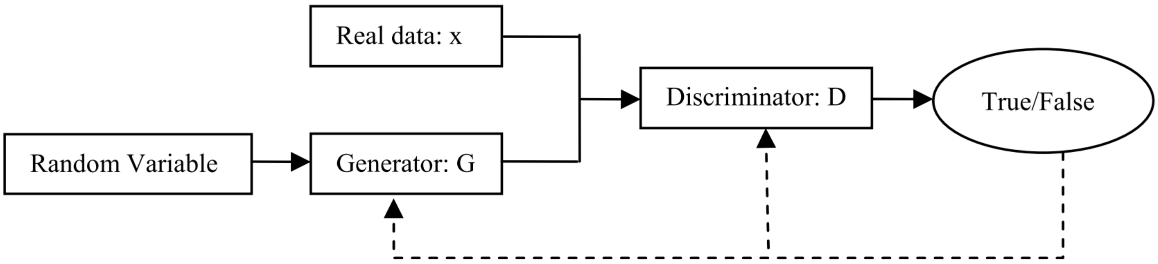


Figure 2.2. GANs Structure Sketch

2.4. LITERATURE STUDY

In a study conducted in 2015, S. Khirade et al. addressed the issue of plant disease identification by employing digital image processing techniques and a back propagation neural network (BPNN) [1]. Various strategies have been developed by researchers to detect plant diseases through the utilization of leaf pictures. The researchers utilized Otsu's thresholding technique in conjunction with border detection and spot detection algorithms to effectively segregate the diseased region within the leaf. Subsequently, the aforementioned properties, including color, texture, morphology, and edges, were extracted to facilitate the classification of plant diseases. The Backpropagation Neural Network (BPNN) is commonly employed in the field of plant disease detection as a classification tool.

In their study, researchers Shiroop Madiwalar and Medha Wyawahare conducted an analysis of several image processing methodologies employed in the identification of plant diseases [2]. The color and linguistic properties were evaluated by the authors in order to detect plant illness. The methods were tested on a dataset consisting of 110 RGB photos. The features utilized for classification encompassed the mean and standard deviation of the RGB and YCbCr channels, as well as the gray level cooccurrence matrix (GLCM) features. Additionally, the mean and standard deviation of the picture coevolution were computed using the Gabor filter. The classification task was performed using support vector machine classifiers. The researchers reached the conclusion that GLCM features demonstrate efficacy in the detection of normal leaves. Color characteristics and Gabor filter traits are well recognized as effective methods for detecting anthracnose infected leaves and leaf spot, respectively. The researchers have successfully attained a peak accuracy of 73.34% by utilizing all of the retrieved attributes.

In their study, Moghadam et al. presented an investigation on the utilization of hyperspectral imaging for the purpose of detecting plant diseases. The research employed the visible and near-infrared (VNIR) as well as the shortwave infrared (SWIR) spectra. The k-means clustering algorithm has been employed by authors for the purpose of leaf segmentation in the spectral domain. A unique technique for grid removal in hyperspectral pictures has been proposed by the researchers. The authors have reported an accuracy of 73% when utilizing vegetation indices within the visible and near-infrared (VNIR) spectral region, while achieving a higher accuracy of 93% when employing the complete spectrum. Despite the increased precision achieved by the suggested method, its implementation necessitates the use of a hyperspectral camera equipped with 324 spectral bands, hence resulting in a significant increase in cost.

In their study, Sharath D. M. et al. devised a detection system for Bacterial Blight in Pomegranate plants. The system included many variables, including color, mean, homogeneity, standard deviation, variance, correlation, entropy, and edges. The authors have successfully employed the grab cut segmentation technique to accurately delineate the area of interest within the given image [4]. The Canny edge detection algorithm was employed for obtaining the borders from the photos. The authors have effectively devised a system capable of accurately forecasting the extent of infection in fruit.

Garima Shrestha et Al. deployed the convolutional neural network to detect the plant disease [5]. Authors have successfully classified 12 plant diseases with 88.80% accuracy. The dataset of 3000 high resolution RGB images were used for experimentation. The network has 3 blocks of convolution and pooling layers. This makes the network computationally expensive. Also the F1 score of the model is 0.12 which is very low because of a higher number of false negative predictions.

In a study conducted by Pranesh Kulkarni et al., titled "Plant Disease Detection Using Image Processing and Machine Learning," the researchers developed a system capable of detecting 20 different diseases in five common plant species with a notable accuracy rate of 93%. The study involved the examination of 25 distinct classes for experimentation. To achieve disease detection, the authors employed the extraction of essential image features, including shape, texture, and color. These features played a crucial role in distinguishing between healthy plants and those affected by various diseases. The study reported an average accuracy of 93% and an F1 score of 0.93, underlining the system's effectiveness. The Random Forest classifier was employed for classification tasks; however, the study acknowledged its susceptibility to overfitting issues. Notably, the system was deemed computationally efficient, ensuring its practicality for real-world deployment in plant disease detection scenarios.

Davinder Singh, et al., in their work "PlantDoc: A Dataset for Visual Plant Disease Detection," introduced the PlantDoc dataset, a valuable resource for plant disease detection research. PlantDoc represents a significant addition to the field, providing a specialized dataset aimed at aiding the development of accurate disease detection models for plants. The authors discussed the performance of MobileNet, achieving a mean Average Precision (mAP) of 22 when evaluated on the COCO dataset, which features a more extensive array of classes. It is worth noting that the PlantDoc dataset contains images that may pose challenges for classification, potentially leading to incorrect categorizations. This highlights the importance of thorough evaluation and the necessity for robust disease detection models in the domain of plant pathology. The availability of the PlantDoc dataset serves as a valuable resource, offering the potential to advance research and practical applications in the fields of agriculture and environmental sciences.

In a study published by Wasswa Shafik et al. in May 2023, presents a comprehensive strategy for data collection and preprocessing in the context of Plant Disease Detection (PDD), catering to both academic and business needs. Notably, the study highlights a limitation in the availability of publicly accessible datasets within this domain. The study underlines the necessity to bridge the gap between controlled laboratory environments and the real-world utilization of DL models in the context of plant disease detection. This paper sheds light on critical aspects of PDD, providing insights into data collection, the limitations of publicly available datasets, and the dominant role of CNN-based systems in addressing plant diseases, particularly in crops like citrus.

In the research conducted by Muhammad E. H. and his team, for the subject of “Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques” the authors explore various versions of the U-net architecture to identify the most effective segmentation model. This determination is achieved through a thorough comparison of the model's predictions with the ground truth segmented images. The results obtained from this investigation demonstrate the superior performance of their model when compared to recent deep learning techniques. Notably, this achievement is accomplished using the widely accessible Plant Village dataset. Muhammad E. H. and his collaborators have thus contributed to the field of leaf disease detection by leveraging deep learning techniques to enhance accuracy and reliability in the identification of plant diseases.

In the paper authored by Monika Lamba and her colleagues, for the subject of “Classification of Plant Diseases Using Machine and Deep Learning” the authors introduce a novel model for plant disease classification. Their model incorporates the Auto-Color Correlogram as an image filter and deploys deep learning (DL) as classifiers with different activation functions. The outcome of their research is remarkable, achieving an accuracy of 99.4% and a sensitivity of 99.9% for binary classification, as well as an accuracy of 99.2% for multiclass classification. Notably, their proposed model surpasses other existing approaches, including LibSVM, SMO, and DL. This study underscores the significance of generating intermediary data representations that other machine learning methods might struggle to achieve. Monika Lamba and her team have made a significant contribution to the field of plant disease classification by providing a high-accuracy model that leverages both machine learning and deep learning techniques.

CHAPTER 3

METHODOLOGY

An "Artificial Intelligence" (AI) system for identifying plant diseases is a multi-step process involving various techniques and technologies. A critical component of this system is the utilization of sensors and cameras to capture images of plant leaves. These images are used to develop and evaluate deep-learning models for disease detection. The initial step involves acquiring plant images through farm-installed cameras. These images undergo pre-processing, including noise reduction, data normalization, and feature normalization, to prepare them for analysis.

The next stage involves plant image analysis to segment the photos and identify areas of interest. Area-based segmentation, based on leaf color, is employed to distinguish healthy and diseased regions. Once the infected areas are accurately identified, the segmentation process continues, and the resulting images are used to train and test deep-learning models. These models employ techniques like convolutional networks, recurrent networks, and support vector machines to classify images as "healthy" or "diseased."

The integration of AI in plant disease detection offers numerous benefits for farmers. It enables accurate and timely identification of diseases, potentially leading to higher plant yields and quality. This approach also reduces the need for manual monitoring and visual inspection, saving time and labor. Moreover, it promotes more sustainable farming practices by reducing reliance on pesticides and chemicals, making agriculture more environmentally friendly.

Combining AI and deep learning for plant disease detection has the potential to revolutionize agriculture. By providing precise and efficient disease detection, farmers can take proactive measures to protect their plants and improve their overall yield and quality. This approach also encourages more sustainable and eco-friendly farming practices, benefiting both farmers and the broader community.

Pre-processing is applied to images sourced from the database and cameras. This involves tasks such as image scaling, enhancement, RGB to grayscale conversion after edge detection, as grayscale images provide the highest accuracy for defect identification. Various analytical approaches are then employed to categorize photos based on specific issues. Images with identified diseases are subsequently uploaded to the cloud using optimization techniques. While real-time pre-processing already exists, its integration into the embedded context is a key goal. This strategy aims to combine deep neural network modeling with modern image processing techniques, allowing for a certain level of automation in regular image capture. This, in turn, enables users to monitor environmental parameters without physically entering the field.

A neural network comprises interconnected artificial neurons with weighted connections, organized in layers based on predefined patterns. Each neuron receives input stimuli, analyzes it, and transmits outputs to other neurons through connected links. Networks primarily rely on learning to adapt to their contexts. Neural networks consist of several components, with the four primary ones being input, node-to-node connectivity, activation function, and learning function. The integration of these components enables a neural network to acquire knowledge effectively.

3.1 DATASET

For this experiment, Sharada P. Mohanty et alPlantVillage .'s public dataset for plant leaf disease identification was employed [19]. The dataset comprises 87 000 RGB photos of healthy and diseased plants split into 38 groups of leaves. We have only chosen 25 classes to determine our method on. Table 1 displays these classes. The researchers decided to concentrate their investigation on 25 of these groups, which represented a variety of plant species and disease kinds. In their study, Table 1 gave a comprehensive analysis of these 25 classifications, detailing the number of pictures in each class and the particular illnesses or disorders they represented.

This dataset allowed the researchers to train and test their deep learning model on a sizable and varied collection of photos, guaranteeing that it could successfully recognize a variety of plant illnesses. They used a deep learning model called CNN, that is particularly effective at picture categorization tasks. Their research outlined an improved deep learning model over traditional machine learning techniques at diagnosing plant illnesses with high levels of accuracy. Their research represents a significant advancement in the use of technology for evaluation and management of plant illness by utilizing strength for deep learning and the sizable dataset provided by PlantVillage.

Characteristics of dataset include:-

1. Sum of 87,000 RGB pictures of both healthy and sick plants are included in the dataset.
2. The photos are divided into 38 distinct leaf groupings that each depict a different plant species or type of illness.
3. For their experiment, the researchers decided to concentrate on 25 of these groups, which were selected to symbolize a range of plant species and disease types.
4. Researchers can test their model on both sorts of images since the dataset contains both color and grayscale photos.
5. A number of sources, including both controlled situations and natural settings, were used to gather the photos in the collection.
6. The dataset provides a variety of viewpoints for the model to learn from, including photographs of both close-up views of individual leaves and larger views of entire plants.
7. The photos in the collection are of different quality and resolution, imitating the actual environmental factors that may be present for diagnosing plant diseases.

3.2 DEEP NEURAL NETWORKS

Each layer of linked nodes or neurons in a DNN processes and refines the incoming data in successive levels. Each connection between a node in a layer and a node in a layer above or below has a weight that affects how strong a signal will be when it goes through that link. The weights of these connections are changed during training to improve the network's performance on a particular task, such as speech or picture categorization. The capacity of DNNs to automatically learn and extract features from the raw input data is one of its main advantages. This indicates that they don't require human feature engineering and can be trained on enormous datasets. At number programmes, including computer vision, processing of natural language, voice detection, and predictive modeling, DNNs have shown to be quite successful. Farmers and other agricultural sector stakeholders can reduce the impact of plant illnesses on plant yields by taking preventive measures by being able to predict the emergence of plant diseases. Predictive modeling, machine learning, and data analysis are all techniques that may be employed to foretell the presence of plant diseases. These techniques entail the analysis of information on the environment, plant health, and other elements that might influence the emergence of plant diseases. However, in order to achieve high levels of accuracy, DNNs need a lot of labeled training data, which can be computationally expensive. Furthermore, it can be challenging to comprehend how DNNs arrive at their predictions or decisions due to the complexity of their internal workings.

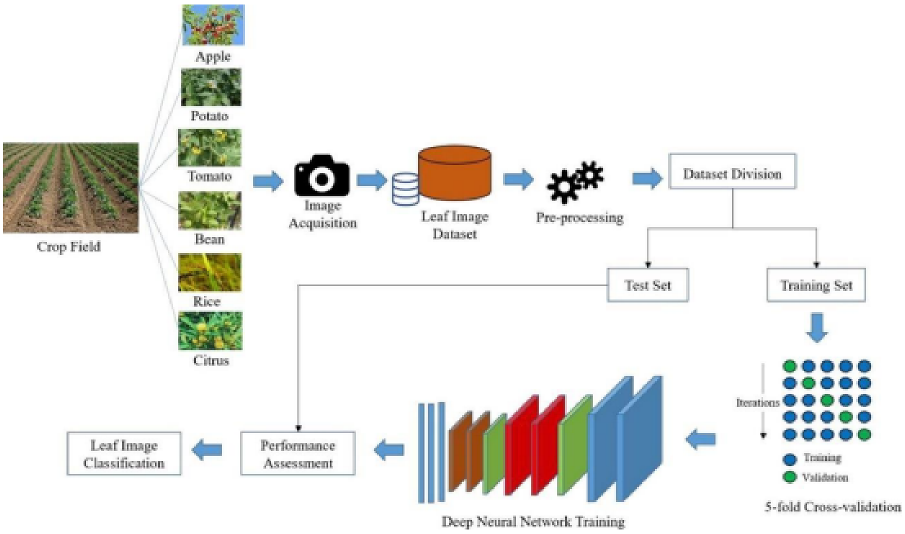


Fig 3.1. Dense convolutional neural networks

Despite these difficulties, the impressive performance of DNNs on a variety of tasks has led to their rising popularity in recent years. They are being employed extensively in research and industry, and it's anticipated that they are going to continue performing a significant part in the development of artificial intelligence.

3.3 RESNET-18

The ResNet series of deep neural networks, which includes ResNet-18, was developed to solve the issue of disappearing gradients in extremely deep networks. The introduction of residual connections, which enables the network to learn residual functions that may be added to the input of each layer, is the main innovation in ResNet18. This makes it easier for gradients to be back propagated across the network, which helps to solve the vanishing gradient problem. Additionally, because of the residual connections, deeper networks can be trained without experiencing performance degradation, which frequently happens with conventional CNNs.

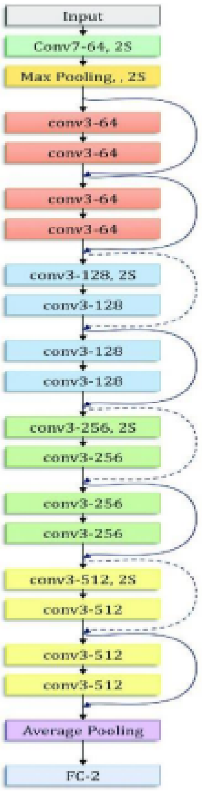


Figure 3.2. ResNet-18 Architecture Diagram

ResNet-18's efficacy in predicting agricultural diseases has also been enhanced by combining it with other methodologies, including transfer learning and data augmentation. For instance, a work that was published in the Journal of Imaging improved ResNet-18 on a dataset of pictures of apple leaves with three distinct illnesses by using transfer learning. The researchers discovered that transfer learning considerably increased ResNet-18's classification accuracy, resulting in a test set accuracy of 98.2%.

3.4 CNN

Christian Szegedy et al. presented the deep neural network architecture known as Inception-v4 in 2016. It is a member of the convolutional neural network family Inception, which is renowned for its efficiency in image categorization tasks. With the addition of various new features, Inception-v4 was created to enhance the functionality of its predecessors, Inception-v1, Inception-v2, and Inception-v3. The utilization of the Inception module, a building component made up of several parallel convolutional layers with various filter sizes, is one of the main aspects of Inception-v4. For tasks like picture identification and classification, this enables the network to collect characteristics of various sizes and complexity.

The addition of user-friendly interfaces that can be customized to suit regional languages and speech-to-text capabilities is another noteworthy feature. This enables farmers to discuss information based on their locality in forums that may be added to the program. The significance of inclusion and accessibility in technological solutions for the agriculture sector is emphasized by this part of the project. To sum up, the project "Plant Disease Prediction Using Artificial and Deep Neural Networks" innovates by combining deep neural networks with A.I technologies to precisely and effectively identify plant illnesses.

By enabling early disease diagnosis, enhancing the sustainability and efficiency of farming operations, and emphasizing accessibility and inclusivity for farmers, the system has the potential to completely transform the agricultural sector. This enables farmers to discuss information based on their locality in forums that may be added to the program. The significance of inclusion and accessibility in technological solutions for the agriculture sector is emphasized by this part of the project.

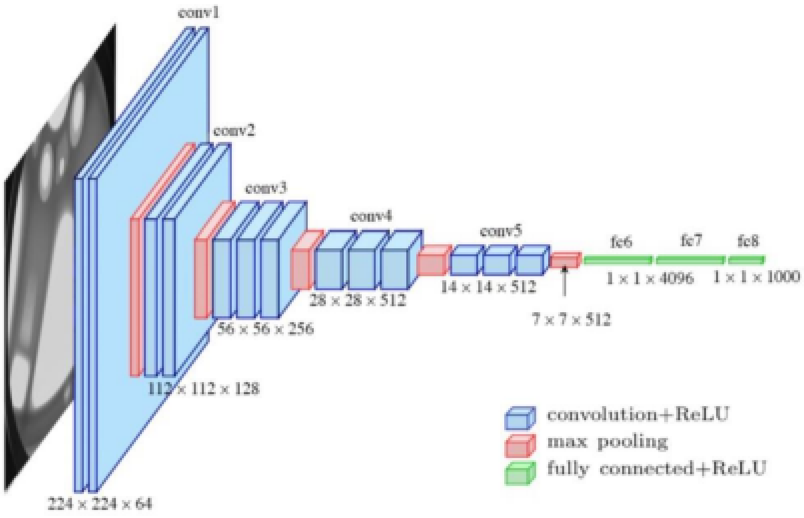


Fig 3.3. Architecture of CNN

3.4.1 Module I(Data collection)

Data gathering is a crucial part of the whole process for plant disease prediction utilizing Artificial Intelligence and deep neural networks. Data on the plants, their diseases, and their growing conditions are collected as part of this process. The accuracy of the prediction models can be considerably impacted by the type and volume of data gathered. For the purpose of predicting agricultural diseases, a variety of data-collecting techniques are available, including crowdsourcing, ground-based sensors, and remote sensing.

In order to collect information on plant health and growth patterns, remote sensing techniques such as satellite images are used. On the other hand, ground-based sensors are positioned right in the field to track environmental variables including humidity, temperature, and soil moisture.

The use of Artificial Intelligence for data gathering for agricultural disease forecasting has become more popular in recent years.

1. To gather information in actual time on elements like climate, moisture, soil moisture, other environmental variables, A.I sensors can be positioned directly on the plants or in the surrounding environment.

2. The development of prediction models using this data can assist farmers in identifying and preventing plant illnesses.

3. Another technique for gathering data that has become more popular recently is crowdsourcing.

4. Data from a large number of people is gathered through crowdsourcing, typically using social media platforms or mobile apps.

5. Using this information, a comprehensive database of plant diseases and their signs can be created.

It's important to ensure that the collected data represents the population being studied and that it is of high quality. Additionally, obtaining informed consent and protecting patient privacy are critical considerations in the data collection process.

Data collecting for plant disease prediction utilizing A.I and deep neural networks is a challenging process that needs careful planning and execution. It is possible to guarantee that the data gathered is reliable and pertinent by utilizing a variety of data sources and cutting-edge technology, which can eventually result in more accurate plant disease prediction and prevention.

Gathering data for plant disease prediction utilizing A.I and deep neural networks is a difficult process that needs careful planning and execution. Utilizing a variety of data sources and cutting-edge technologies can aid in ensuring that the information gathered is accurate and pertinent, which can ultimately result in more accurate plant disease prediction and prevention.

Figure 3.4 Data Collection Representation

3.4.2 MODULE II(FEATURE EXTRACTION)

Once you have the data, you need to preprocess it to prepare it for use in your model. To assess the effectiveness of your model, it’s important to divide your data into three groups: a validation set, training set or test set. DDSM images vary in gray level due to differences in the scanners used to create them. Likewise, the images have varying optical densities and need to be normalized before the training process can begin.

Feature extraction is a crucial step that involves identifying appropriate components in the bare dataset to as worked as input on the deep learning model. Feature extraction is important because raw data such as medical images can be high dimensional and contain a lot of noise, making it difficult for deep learning models to learn meaningful patterns directly from the data. Feature extraction can be performed using different techniques, including:

1. CNN : CNNs are frequently employed in picture-based classification applications for feature extraction. They are built to automatically recognize patterns in the input data and extract pertinent features.

2. Pretrained Models: pretrained models such as VGG, ResNet, and Inception are trained on large image datasets and can be used for feature extraction by removing the last few layers of the model and using the output from the remaining layers as features.

3. Handcrafted Features: Handcrafted features can be taken from raw data utilizing processing of images methods, such as edges detect or texture analysis.

Once the features are extracted, they are typically fed into a deep-learning model for classification. To know what to choose as feature extraction, the technique depends on the nature of the data, difficulty of classification work, or availability of labeled data for training.

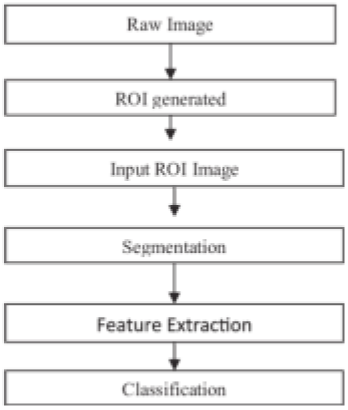


Figure 3.5 Feature Extraction Representation

3.4.3 MODULE III (SELECTING AND CREATING MODEL)

The next is selecting the model based on the hyperparameter, type of layer, size, etc. The techniques above mentioned will help to design and build a model for better classification and prediction. After feature extraction in Plant disease classification using deep learning, the following step is to choose a suitable deep learning model of classification task. Choice of model is reliant on various substitutes, including the nature of the dataset, difficulty for classification work, number of classes, the accessibility of labeled data. Some common types of deep learning models that possibly used for Plant disease classification include:

1. CNN: CNNs are commonly used for image-based classification tasks, including Plant disease classification. They are particularly effective in capturing spatial dependencies and extracting relevant characteristics of the input dataset.

2. RNN: RNNs commonly employed for sequential data classification tasks, such as time-series analysis or natural language processing. They can also be used for Plant disease classification using features extracted from sequential data, such as gene expression data.

3. Ensemble Models: Ensemble models combine multiple deep learning models to enhance accuracy or robustness of classification task. This can be particularly effective when dealing with complex or noisy data.

The selection of the appropriate deep learning model should be based on effectiveness of model throughout verification set the training procedure. It is crucial to make sure that the chosen model is capable of achieving high accuracy on the testing set and that it can generalize well to new data.

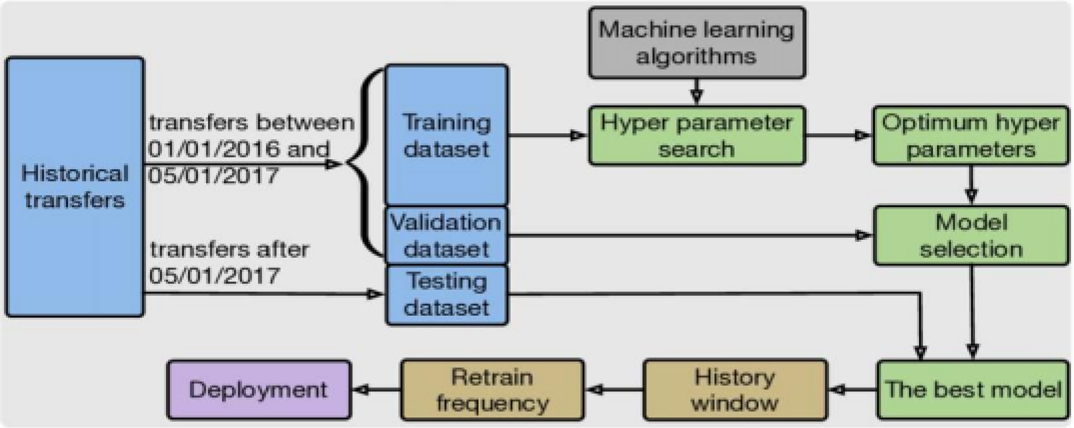


Figure 3.6: Model Selection Flowchart

3.4.4 Module IV (Multiple layers of classification)

This Module is the new thing that we added in this project. In these the original two classifications that are Benign and Malignant are further divided in four sub-layers each for all the classification of the tissue image that we extracted from the dataset. Multiple layers of classification can be used after selecting a model in Plant disease classification using deep learning. This involves using multiple deep learning models or layers to perform hierarchical classification of the input data. For example, in a Plant disease classification task, the first layer of classification may involve distinguishing between benign and malignant tumors. Multiple layers of classification can be implemented using various techniques, including:

1. Stacked Ensemble Models: Stacked ensemble models combine multiple deep learning models, each trained on a different subset of features or input data, to perform hierarchical classification.

2. Multi-Task Learning: A single deep learning model is trained to carry out several categorization tasks at once using multi-task learning. In light of Plant disease classification, this could involve training a single model to distinguish between benign and malignant tumors as well as classify malignant tumors into subtypes.

3. Transfer Learning: Learning Transfer entails utilizing a pre trained deep learning model as feature extractor for fresh classification work. In light of classification, this could involve using a pretrained model trained on a similar classification task, such as lung tumor classification, to extract relevant features for the Plant disease classification task.

The choice of multiple layers of classification technique depends on difficulty of classification work, number of classes, accessibility of labeled dataset for training. It is important to evaluate the performance of the multiple layers of classification approach on the validation and testing sets to ensure that it is effective in improving the accuracy of the classification task.

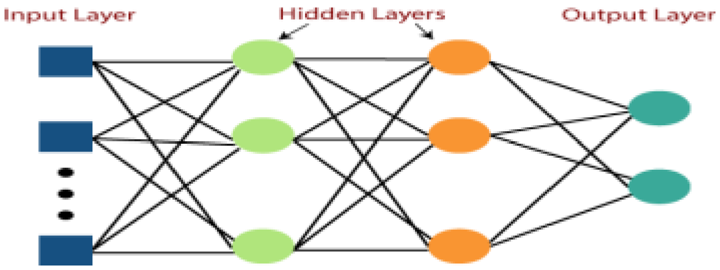


Figure 3.7 Multiple Layers of Classification

3.4.5 Module V (Training and Evaluation)

Upon the completion of the model architecture and compilation, you can start training the model with a training set. Evaluation of its effectiveness is based on validation dataset. This allows you to tune the hyperparameters and make any necessary modifications to the model architecture. After implementing multiple layers of classification in Plant disease classification using deep learning, the following step is training and evaluating our model.

Training the model involves using a labeled dataset to optimize the parameters of utilizing a deep learning model suitable optimization algorithm, such as Stochastic Gradient Descent (SGD). The dataset is typically split into training, validation, and testing sets. The validation set is used to adjust the hyperparameters and avoid overfitting, the training set is used to update the model parameters, and the testing set is used to assess the trained model's performance.

Evaluation of trained models involves calculating performance indicators such as accuracy, precision, recall, F1 score, and ROC curve on the testing set. These metrics provide an estimate of how well the model can classify new, unseen data. In the context of multiple layers of classification in Plant disease classification, the evaluation process can be performed separately for each layer of classification. For example, the accuracy of the first layer of classification can be evaluated by comparing the predicted labels of the benign and malignant tumors to their ground truth labels.

Similarly, the accuracy of the second layer of classification can be evaluated by comparing the predicted labels of the malignant tumor subtypes to their ground truth labels. It is important to ensure that the model(s) perform well on both the validation and testing sets and that they can generalize well to new, unseen data. If the model(s) do not perform well, it may be necessary to retrain the model(s) with different hyperparameters or adjust the architecture of the deep learning model(s).

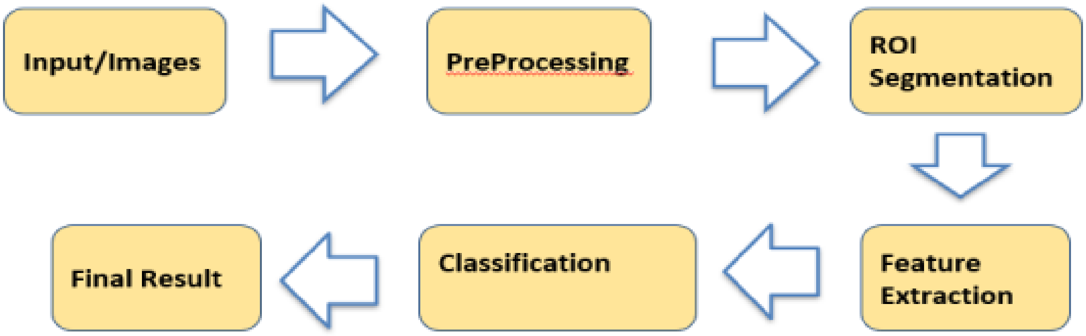


Figure 3.8 System Architecture Diagram

After implementing multiple layers of classification using deep learning, the following step is training and evaluating our model. Training the model involves using a labeled dataset to optimize the parameters of the deep learning model using a suitable optimization algorithm, such as Stochastic Gradient Descent (SGD). The dataset is often separated into testing, validation, and training sets. The training set is used to update the model parameters, the testing set is used to evaluate the performance of the trained model, and the validation set is used to correct the hyperparameters and avoid overfitting.

Utilizing a deep learning model that has already been trained is known as transfer learning model as feature extractor for a new classification task. In light of classification, this could involve using a previously trained model skilled in a comparable classification task, such as classification, to extract relevant features for the classification task.

CHAPTER 4

RESULT AND DISCUSSION

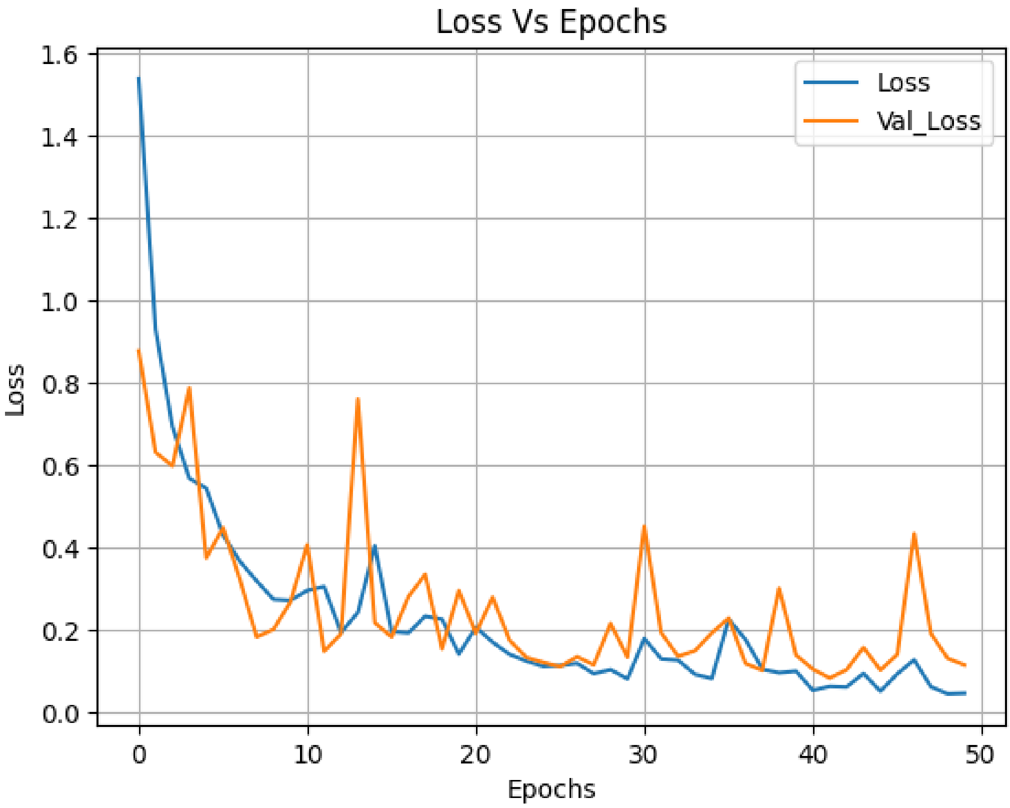


Fig. 4.1 Data Accuracy Graph

The amount and consistency of the practice data, the complexity of prototype architecture, or choice of hyperparameters can all affect how accurate a deep learning model is at classifying Plant diseases. The dataset must be properly chosen and preprocessed, the model architecture must be adequate, and the model hyperparameters must be optimized using methods like cross-validation and grid search. In order to make sure that the model generalizes effectively to previously unexplored data, it is also critical to assess the model's performance on a validation dataset.

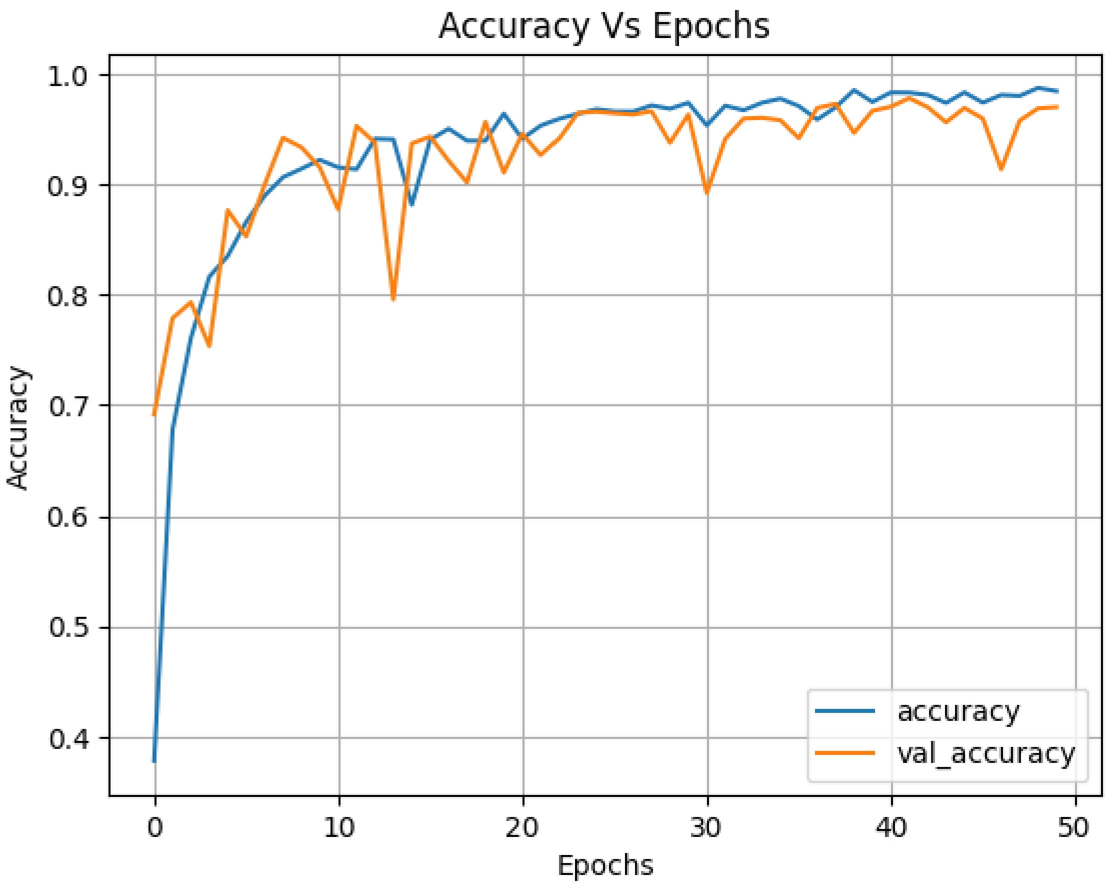


Fig. 4.2 accuracy V/S Epochs

The term "epochs" refers to the number of complete passes that a machine learning model, such as a DNN, makes through the entire training dataset. The interplay between accuracy and epochs has a profound impact on the effectiveness of disease detection.

During the training process, the DNN learns to recognize patterns and features within the dataset, enabling it to make accurate predictions about whether a leaf is healthy or afflicted by a disease. Typically, as the number of epochs increases, the DNN's accuracy on the training data also improves. This is because the network refines its internal parameters and weights to better fit the training data. To find the optimal balance between accuracy and epochs, it is common practice to use a validation dataset. This dataset, separate from the training data, allows for the continuous monitoring of the DNN's accuracy as it is trained over multiple epochs. Ideally, the validation accuracy should also improve as epochs increase, but when it begins to decrease or stabilize, it signals that the model is overfitting.

In order to increase model accuracy, additional information may be gathered in addition to photos, such as weather data, soil quality data, and other environmental parameters. A.I gadgets like weather sensors, moisture sensors, and other environmental monitoring tools may be used to do this. In order to train precise and dependable models, data collection for plant disease prediction utilizing A.I and deep neural networks entails the capture, labeling, and integration of many forms of data. Data collection is a crucial stage in the 44 creation of efficient systems for the prediction of plant diseases since the quality and amount of the data gathered directly affects the performance of the models

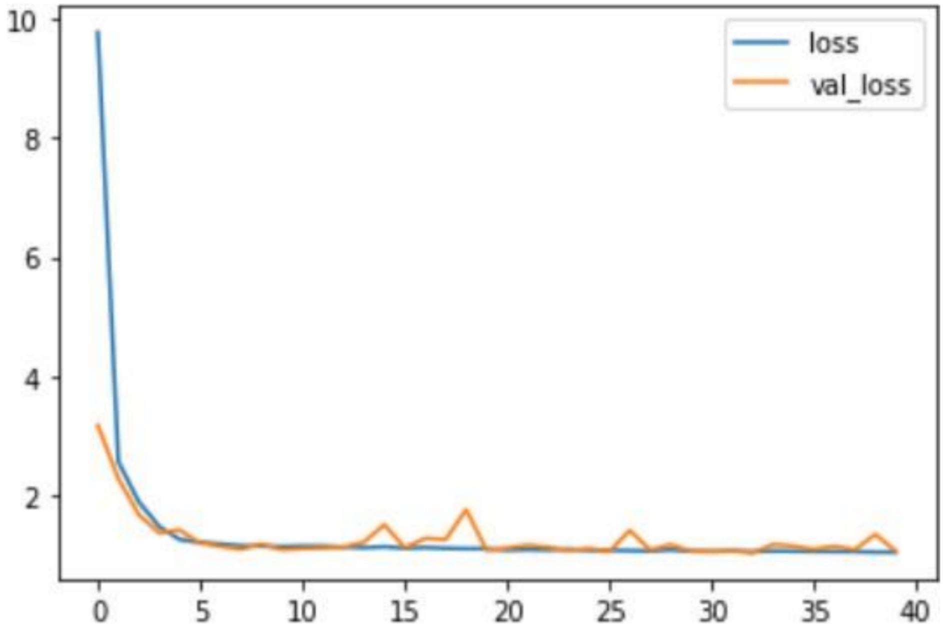


Fig. 4.3 Error Loss Graph

During the training of a deep learning model for Plant disease classification, the model undergoes multiple iterations or epochs, where it learns to identify patterns and features in the images. In each epoch, the model updates its weights the discrepancy between the anticipated and actual values, or loss function, must be reduced in order to actual label of the images. To evaluate the model's performance during training, it is crucial to monitor both the inaccuracy in training and the validation error. The learning blunder measures difference between the predicted and actual labels on the training data, while the validation error measures the error on a separate set of data that the model has not seen before. If the low training error, and the validation high error, it that the model is too closely fitting the training set of data and may not generalize well to new data. Conversely, if both the training error and validation error are high, it implies that the model is underfitting and not learning enough based on the data. To enhance model's efficiency, the goal is to minimize the validation error while avoiding overfitting. This can be accomplished by changing the model's hyperparameters, including the learning rate, the number of layers, and the amount of neurons, and through regularization techniques like dropping out and shed pounds. By monitoring training and validation errors during training, healthcare providers can ensure that the deep learning model is accurately and reliably classifying Plant diseases, leading to better treatment.

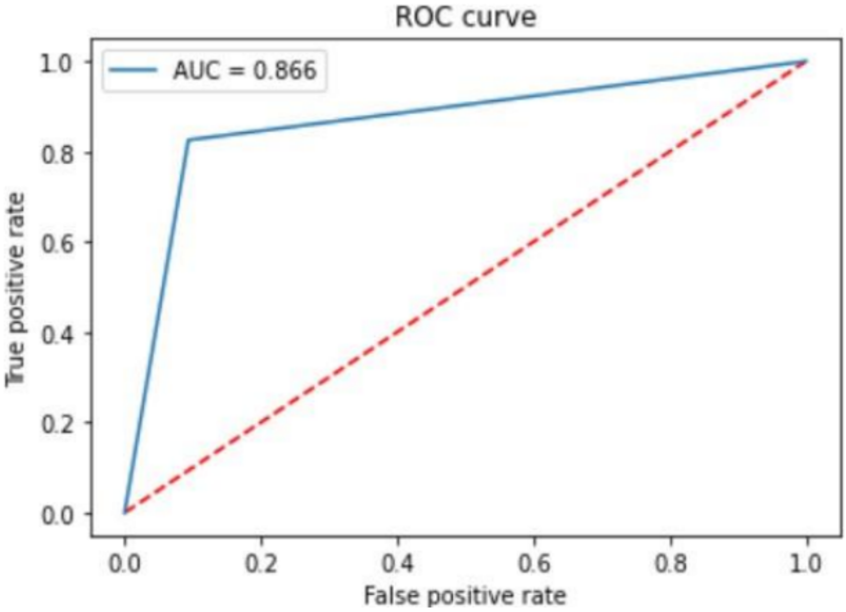


Fig. 4.4 ROC Performance curve

Crop disease prediction using deep learning can be evaluated using a ROC bend, which shows the performance of the binary classifier as the discrimination threshold is changed graphically. To generate a ROC curve, the deep learning model is first built from a dataset of breast tumor pictures with labels indicating whether each tumor is benign or malignant. Then, the model is applied to a separate validation dataset, and the predicted probabilities of each tumor being malignant are calculated. A range of classification thresholds is chosen, and the true positive rate and false positive rate is calculated for each threshold. The TPR represents the proportion of malignant tumors correctly classified as such, while the FPR represents the proportion of benign tumors incorrectly classified as malignant. These values are plotted on a graph where the x-axis is the FPR and the y-axis is the TPR, and the ROC curve for the deep learning model is plotted on top of a diagonal line that represents the ROC curve for a random classifier. The AUC of the ROC curve can be used as a measure of the overall performance of the deep learning model, with 1.0 AUC indicating perfection classification performance with an AUC is 0.5 indicating random guessing. By evaluating the ROC curve and AUC, healthcare providers can assess the accuracy and reliability of the deep learning model in breast tumor classification, leading to better treatment decisions and improved patient outcomes.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENTS

A Comprehensive Study on Leaf Disease Prediction Using Artificial Intelligence and Deep Neural Networks represents a significant advancement in the field of agriculture, particularly in India, where agriculture plays a crucial role in the livelihoods of millions. This project addresses the challenges of plant disease identification, offering a practical and innovative solution that holds the potential to revolutionize farming practices. Here are the key takeaways from this project:

1. Innovative Solution: The project introduces an innovative solution that leverages Artificial Intelligence and deep neural networks for precise and early identification of plant diseases. It overcomes the limitations of manual inspection methods, which are labor-intensive, time-consuming, and often less accurate.

2. Early Disease Detection: One of the primary benefits of this project is its capability for early disease detection. By using AI-based sensors and deep neural networks to monitor plant health continuously, farmers can identify disease symptoms at their initial stages. This early detection enables them to take prompt action to prevent disease spread and minimize plant losses.

3. Precision Agriculture: The project aligns with the concept of precision agriculture, where data-driven methods are used to optimize plant output while minimizing resource waste. Farmers can make informed decisions about irrigation, fertilization, and other inputs based on real-time data. This not only increases plant yield but also reduces resource wastage.

4. Sustainable Farming: Plant disease prediction using AI and deep neural networks contributes to sustainable agriculture. By reducing the need for pesticides and other chemicals, the project promotes environmentally friendly farming practices. It also supports the well-being of agricultural workers and the local population by

minimizing the health risks associated with chemical exposure.

5. Improved Efficiency: The project significantly improves the efficiency of farming operations. It reduces the need for labor-intensive disease inspections and provides farmers with fast and accurate information for decision-making. This efficiency not only saves time but also reduces operational costs.

6. User-Friendly Interface: The project focuses on user-friendliness, incorporating farmer forums for localized discussions and supporting local languages. This ensures that farmers, regardless of their tech-savviness, can access essential information about their plant health and take timely action to prevent disease spread.

7. Economic Viability: By minimizing plant losses and reducing resource waste, the project supports the economic viability of farming operations. It enables farmers to increase their yields and revenues, ultimately contributing to their economic well-being.

A Comprehensive Study on Leaf Disease Prediction Using Artificial Intelligence and Deep Neural Networks is a significant step toward sustainable and efficient farming practices in India and beyond. It harnesses the power of AI and deep learning to address the challenges of plant disease identification, offering a practical and effective solution that benefits farmers, the environment, and food security. This project has the potential to transform the agricultural sector and enhance the livelihoods of millions of farmers.

CHAPTER 6

REFERENCES

[1] Hernández, E.; Sanchez-Anguix, V.; Julian, V.; Palanca, J.; Duque, N. Rainfall prediction: A deep learning approach. In International Conference on Hybrid Artificial Intelligence Systems; Springer: Cham, Switzerland, 2016; pp. 151–162.

[2] Goswami, B.N. The challenge of weather prediction. Resonance **1996**, 1, 8–17.

[3] Nayak, D.R.; Mahapatra, A.; Mishra, P. A survey on rainfall prediction using artificial neural network. Int. J. Comput. Appl. **2013**, 72, 16.

[4] Kashiwao, T.; Nakayama, K.; Ando, S.; Ikeda, K.; Lee, M.; Bahadori, A. A neural network-based local rainfall prediction system using meteorological data on the internet: A case study using data from the Japan meteorological agency. Appl. Soft Comput. **2017**, 56, 317–330.

[5] Mislan, H.; Hardwinarto, S.; Sumaryono, M.A. Rainfall monthly prediction based on artificial neural network: A case study in Tenggarong Station, East Kalimantan, Indonesia. Procedia Comput. Sci. **2015**, 59, 142–151.

[6] Muka, Z.; Maraj, E.; Kuka, S. Rainfall prediction using fuzzy logic. Int. J. Innov. Sci. Eng. Technol. **2017**, 4, 1–5.

[7] Nikesh Gondchawar, Dr. R. S. Kawitkar, "A.I based Smart Agriculture", IJARCCE, Vol. 5, Issue 6, June 2016

[8] Zarreen Naowal Reza, Faiza Nuzhat, Nuzhat Ashraf Mahsa, Md. Haider Ali , "Detecting jute plant disease using image processing and machine learning", 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT).

[9] Meliks Ardogan, Adam Tuncer, Yunnus Ozen, "Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm", 2018 3rd International Conference on Computer Science and Engineering (UBMK).

[10] Jie You, Wei Liu, Joonwhoan Lee, “A DNN-based semantic segmentation for detecting weed and plant”, Computers and Electronics in Agriculture, Vol. 178, November 2020.

[11] Yuzhen Lu, Sierra Young, “A survey of public datasets for computer vision tasks in precision agriculture, Computers and Electronics in Agriculture,”, Volume 178, 2020, 105760, ISSN 0168- 1699.

[12] S. Ramesh and Bharghava Rajaram,” A.I Based plant Disease Identification System Using Optimization Techniques”, ARPN Journal of Engineering and Applied Sciences vol. 13, no.4, february 2019.

[13] Pallavi. S. Marathe, “Plant Disease Detection using Digital Image Processing and GSM”, International Journal of Engineering Science and Computing, April 2017, pp. 10513-15.

[14] R.Rajmohan, M.Pajany, Smart paddy plant disease identification and management using deep convolution neural network & svm classifier, International journal of pure and applied mathematics, vol 118, no 5, pp. 255-264, 2017.

[15] A. A. Joshi and B. D. Jadhav. 2016. Monitoring and controlling rice diseases using Image processing techniques. International Conference on Computing. Analytics and Security Trends (CAST), Pune. pp. 471-476.

[16] S. Mutalib, M. H. Abdullah, S. Abdul-Rahman and Z.A. Aziz. 2016. A brief study on paddy applications with image processing and proposed architecture. IEEE Conference on Systems, Process and Control (ICSPC), Melaka, Malaysia. pp. 124-129.

[17] Jagadeesh D. Pujari, Rajesh Yakkundimath, Abdulmunaf S. Byadgi. 2015. Image Processing

[18] Abhirami, K., Praba, G.C. and Lakshmi, R.S. (2021) A.II based Paddy plant Disease Identification and prevention system using deep neural networks and image processing, International Journal of 52 Engineering Research & Technology. IJERT-International jurnal of Engineering Research & Technology. Available at: https://www.ijert.org/A.I-based-paddy-plant-disease-identification-andprevention-system-using-deep-neural-networks-and-image-processing.

[19] Kulkarni, P. et al. (2021) Plant disease detection using image processing and machine learning, arXiv.org. Available at: https://arxiv.org/abs/2106.10698.

[20] Mohanty SP, Hughes DP and Salathé M (2016) Using Deep Learning for Image-Based Plant Disease Detection. Front. Plant Sci. 7:1419. doi: 10.3389/fpls.2016.01419

[21] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image

Classification," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.

[22] Jie You, Wei Liu, Joonwhoan Lee, "A DNN-based semantic segmentation for detecting weed and plant", Computers and Electronics in Agriculture, Vol. 178, November 2020.

[23] Kulkarni, Pranesh & Karwande, Atharva & Kolhe, Tejas & Kamble, Soham & Joshi, Akshay & Wyawahare, Medha. (2021). Plant Disease Detection Using Image Processing and Machine Learning.

[24] Davinder Singh, Naman Jain, Pranjali Jain, Pratik Kayal, Sudhakar Kumawat, and Nipun Batra. 2020. PlantDoc: A Dataset for Visual Plant Disease Detection. In Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (CoDS COMAD 2020). Association for Computing Machinery, New York, NY, USA, 249–253. https://doi.org/10.1145/3371158.3371196

[25] W. Shafik, A. Tufail, A. Namoun, L. C. De Silva and R. A. A. H. M. Apong, "A Systematic Literature Review on Plant Disease Detection: Motivations, Classification Techniques, Datasets, Challenges, and Future Trends," in IEEE Access, vol. 11, pp. 59174-59203, 2023, doi: 10.1109/ACCESS.2023.3284760.

[26] Chowdhury, Muhammad E. H., Tawsifur Rahman, Amith Khandakar, Mohamed Arselene Ayari, Aftab Ullah Khan, Muhammad Salman Khan, Nasser Al-Emadi, Mamun Bin Ibne Reaz, Mohammad Tariqul Islam, and Sawal Hamid Md Ali. 2021. "Automatic and Reliable Leaf Disease Detection Using Deep Learning Techniques" AgriEngineering 3, no. 2: 294-312. https://doi.org/10.3390/agriengineering3020020

[27] Lamba, Monika & Gigras, Yogita & Dhull, Anuradha. (2021). Classification of plant diseases using machine and deep learning. Open Computer Science. 11. 491-508. 10.1515/comp-2020-0122.

[28] Mohanty SP, Hughes DP, Salathé M. Using Deep Learning for Image-Based Plant Disease Detection. Front Plant Sci. 2016 Sep 22;7:1419. doi: 10.3389/fpls.2016.01419. PMID: 27713752; PMCID: PMC5032846.

[29] Shoaib M, Shah B, Ei-Sappagh S, Ali A, Ullah A, Alenezi F, Gechev T, Hussain T, Ali F. An advanced deep learning models-based plant disease detection: A review of recent research. Front Plant Sci. 2023 Mar 21;14:1158933. doi: 10.3389/fpls.2023.1158933. Erratum in: Front Plant Sci. 2023 Sep 27;14:1282443. PMID: 37025141; PMCID: PMC10070872.

[30] J. Kolli, D. M. Vamsi and V. M. Manikandan, "Plant Disease Detection using Convolutional Neural Network," 2021 IEEE Bombay Section Signature Conference (IBSSC), Gwalior, India, 2021, pp. 1-6, doi: 10.1109/IBSSC53889.2021.9673493.

[31] S. Barburiceanu, S. Meza, B. Orza, R. Malutan and R. Terebes, "Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture," in IEEE Access, vol. 9, pp. 160085-160103, 2021, doi: 10.1109/ACCESS.2021.3131002.

[32] V. K. Vishnoi, K. Kumar, B. Kumar, S. Mohan and A. A. Khan, "Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network," in IEEE Access, vol. 11, pp. 6594-6609, 2023, doi: 10.1109/ACCESS.2022.3232917.

[33] K. Roy et al., "Detection of Tomato Leaf Diseases for Agro-Based Industries Using Novel PCA DeepNet," in IEEE Access, vol. 11, pp. 14983-15001, 2023, doi: 10.1109/ACCESS.2023.3244499.

[34] S. Ahmed, M. B. Hasan, T. Ahmed, M. R. K. Sony and M. H. Kabir, "Less is More: Lighter and Faster Deep Neural Architecture for Tomato Leaf Disease Classification," in IEEE Access, vol. 10, pp. 68868-68884, 2022, doi: 10.1109/ACCESS.2022.3187203.

[35] K. P. Asha Rani and S. Gowrishankar, "Pathogen-Based Classification of Plant Diseases: A Deep Transfer Learning Approach for Intelligent Support Systems," in IEEE Access, vol. 11, pp. 64476-64493, 2023, doi: 10.1109/ACCESS.2023.3284680.

[36] L. Li, S. Zhang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," in IEEE Access, vol. 9, pp. 56683-56698, 2021, doi: 10.1109/ACCESS.2021.3069646.

[37] A. C. Cruz, A. Luvisi, L. De Bellis, and Y. Ampatzidis, ‘‘Vision-based plant disease detection system using transfer and deep learning,’’ in Proc. Asabe Annu. Int. Meeting, Spokane, WA, USA, Jul. 2017, pp. 16–19.

[38] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, ‘‘Deep interpretable architecture for plant diseases classification,’’ in Proc. Signal Process., Algorithms, Archit., Arrangements, Appl. (SPA), Poznan, Poland, Sep. 2019, pp. 111–116.

[39] C. DeChant, T. Wiesner-Hanks, S. Chen, E. L. Stewart, J. Yosinski, M. A. Gore, R. J. Nelson, and H. Lipson, ‘‘Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning,’’ Phytopathology, vol. 107, no. 11, pp. 1426–1432, Nov. 2017.

[40] J. Lu, J. Hu, G. Zhao, F. Mei, and C. Zhang, ‘‘An in-field automatic wheat disease diagnosis system,’’ Comput. Electron. Agricult., vol. 142, pp. 369–379, Nov. 2017.

[41] J. G. Ha, H. Moon, J. T. Kwak, S. I. Hassan, M. Dang, O. N. Lee, and H. Y. Park, ‘‘Deep convolutional neural network for classifying Fusarium wilt of radish from unmanned aerial vehicles,’’ J. Appl. Remote Sens., vol. 11, no. 4, p. 1, Dec. 2017.

[42] J. G. A. Barbedo, ‘‘Plant disease identification from individual lesions and spots using deep learning,’’ Biosyst. Eng., vol. 180, pp. 96–107, Apr. 2019.

[43] S. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar, ‘‘An explainable deep machine vision framework for plant stress phenotyping,’’ Proc. Nat. Acad. Sci. USA, vol. 115, no. 18, pp. 4613–4618, Apr. 2018.

[44] Y. Lu, S. Yi, N. Zeng, Y. Liu, and Y. Zhang, ‘‘Identification of rice diseases using deep convolutional neural networks,’’ Neurocomputing, vol. 267, pp. 378–384, Dec. 2017.

[45] A. Picon, A. Alvarez-Gila, M. Seitz, A. Ortiz-Barredo, J. Echazarra, and A. Johannes, ‘‘Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild,’’ Comput. Electron. Agricult., vol. 161, pp. 280–290, Jun. 2019.

[46] A. Johannes, A. Picon, A. Alvarez-Gila, J. Echazarra, S. Rodriguez-Vaamonde, A. D. Navajas, and A. Ortiz-Barredo, ‘‘Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case,’’ Comput. Electron. Agricult., vol. 138, pp. 200–209, Jun. 2017.

[47] M. A. Khan, T. Akram, M. Sharif, M. Awais, K. Javed, H. Ali, and T. Saba, ‘‘CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features,’’ Comput. Electron. Agricult., vol. 155, pp. 220–236, Dec. 2018.

[48] M. Kerkech, A. Hafiane, and R. Canals, ‘‘Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images,’’ Comput. Electron. Agricult., vol. 155, pp. 237–243, Dec. 2018.

[49] K. Nagasubramanian, A. K. Singh, A. Singh, S. Sarkar, and B. Ganapathysubramanian, ‘‘Usefulness of interpretability methods to explain deep learning based plant stress phenotyping,’’ Comput. Sci., vol. 4, pp. 18–32, Jul. 2020.

[50] Y. Sun, Z. Jiang, L. Zhang, W. Dong, and Y. Rao, ‘‘SLIC_SVM based leaf diseases saliency map extraction of tea plant,’’ Comput. Electron. Agricult., vol. 157, pp. 102–109, Feb. 2019.

[51] Z.-W. Hu, H. Yang, J.-M. Huang, and Q.-Q. Xie, ‘‘Fine-grained tomato disease recognition based on attention residual mechanism,’’ J. South China Agricult. Univ., vol. 40, no. 6, pp. 124–132, Jul. 2019.

[52] S. H. Lee, H. Goëau, P. Bonnet, and A. Joly, “New perspectives on plant disease characterization based on deep learning,” Comput. Electron. Agricult., vol. 170, Mar. 2020, Art. no. 105220.

APPENDIX I

In this section of the report, we provide details about the programming languages and packages used to implement our project. The entire project was developed using Python, a versatile, interpreted, interactive, and high-level programming language. Python is known for its readability and clear syntax, making it an ideal choice for implementing complex algorithms in artificial intelligence (AI) and machine learning (ML). The language's flexibility and extensive libraries have enabled developers to create robust AI systems. The following Python packages were employed in our project:

1. **Pandas**: Pandas is a Python package that facilitates the manipulation of structured data. It provides data structures like DataFrames, designed for data analysis. Its user-friendly approach simplifies data handling and analysis, making it a key tool in our project.

2. **NumPy**: NumPy is a fundamental package for scientific computing in Python. It offers support for multi-dimensional arrays and various mathematical functions, essential for efficient array operations, statistics, and linear algebra computations.

3. **OpenCV-Python**: OpenCV is a versatile library for computer vision and machine learning applications. It offers over 2500 optimized algorithms, covering a wide range of computer vision tasks, including object recognition, motion tracking, and image processing.

4. **Imgaug**: Imgaug is a library designed for machine learning experiments that involve image augmentation. It provides a variety of augmentation techniques, simplifying their application on different types of data, including images, key points, bounding boxes, heatmaps, and segmentation maps.

5. **PIL (Python Imaging Library)**: PIL is a library that enhances the image processing capabilities of Python. It is used for efficient manipulation of image data in various pixel formats.
6. **Matplotlib**: Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is particularly valuable for generating plots and charts, which can be incorporated into various applications.

7. **Keras**: Keras is a high-level neural networks API, allowing for the easy construction and training of artificial neural networks. In our project, it is used to interface with the TensorFlow library, providing flexibility and simplicity in neural network development.

8. **Glob**: The Glob module in Python is used for path pattern matching, allowing the program to locate files and directories based on specific patterns. It simplifies tasks such as finding files with specific extensions.

9. **JSON (JavaScript Object Notation)**: The JSON module in Python is used for encoding and decoding JSON data. It simplifies the interaction with JSON, a widely used data format for data exchange between applications.

10. **OS (Operating System)**: The OS module in Python provides tools for working with operating system-related tasks, such as file system operations, directory management, and environment variables. It enables a portable way to handle various file system and OS-specific operations.

11. **Imutils**: Imutils is a package designed to simplify working with OpenCV, a popular computer vision library. It includes functions for operations like color conversions, image display, rotation, cropping, and resizing.

12. **TensorFlow**: TensorFlow is a powerful open-source machine learning and AI toolkit. It is widely used for deep learning tasks and provides a flexible ecosystem for developing and deploying machine learning models.

13. Paths: Paths, in Python, represent the location of files or directories in the file system. The OS module offers functions like join, basename, dirname, abspath, split, splitext, and more for managing file paths.
14. **Scikit-learn (Sklearn)**: Scikit-learn is a free machine learning library that provides a wide range of algorithms for classification, regression, and clustering tasks. It is an essential tool for data analysis and machine learning.

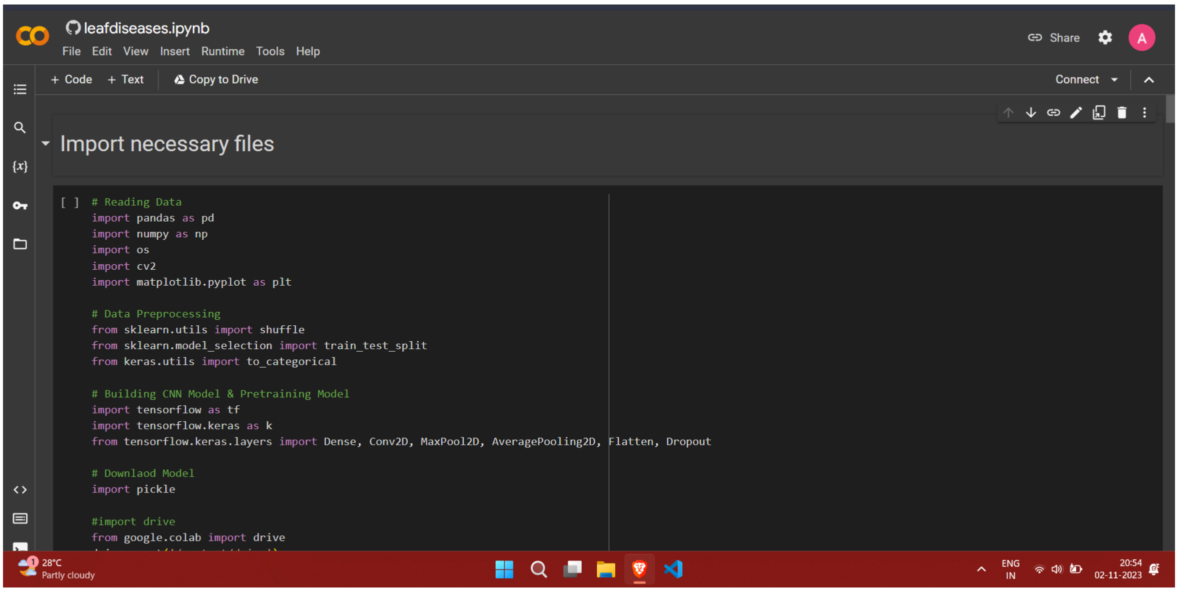
15. **SciPy**: SciPy is an open-source library for scientific and technical computing. It includes modules for signal and image processing, interpolation, integration, special functions, and optimization, making it valuable for scientific computing.

16. **functools**:The functools module offers higher-order functions for modifying and combining existing functions. It includes functions like `partial` for creating new functions with some fixed arguments, `reduce` for aggregating a sequence of values, and `cached_property` for caching property results.

17. **itertools**:The itertools module provides tools for working with iterable objects, such as lists, tuples, and dictionaries. It includes functions for creating and processing iterators efficiently.

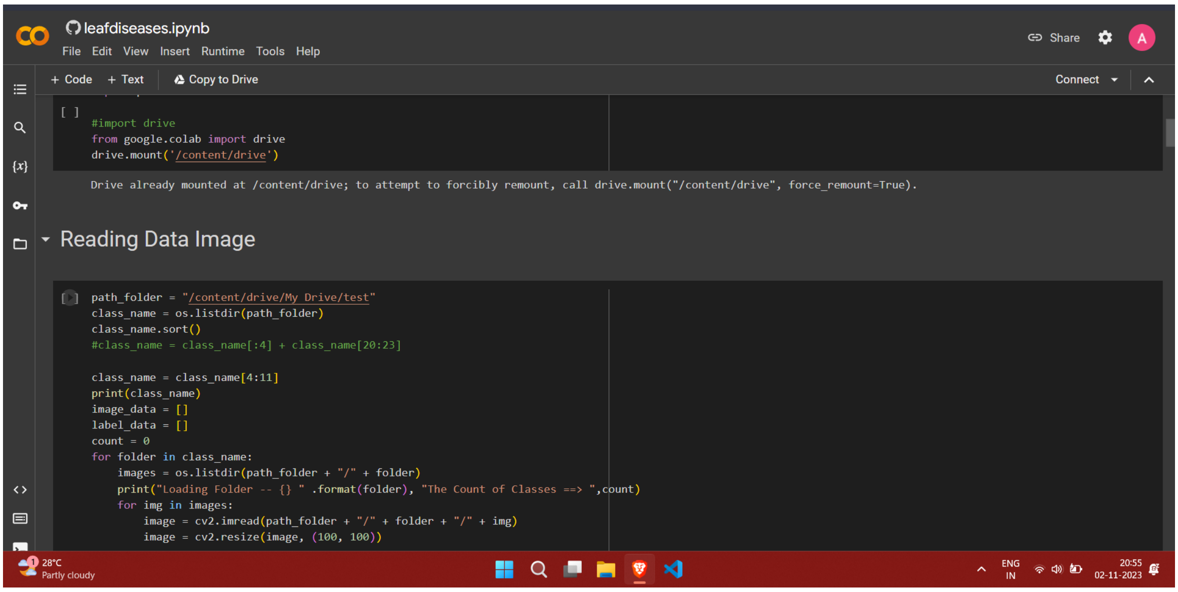
APPENDIX II

This section of the report contains the code we built for our project.



Import necessary files

```
# Reading Data
import pandas as pd
import numpy as np
import os
import cv2
import matplotlib.pyplot as plt

# Data Preprocessing
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical

# Building CNN Model & Pretraining Model
import tensorflow as tf
import tensorflow.keras as k
from tensorflow.keras.layers import Dense, Conv2D, MaxPool2D, AveragePooling2D, Flatten, Dropout

# Downlaod Model
import pickle

#import drive
from google.colab import drive
```

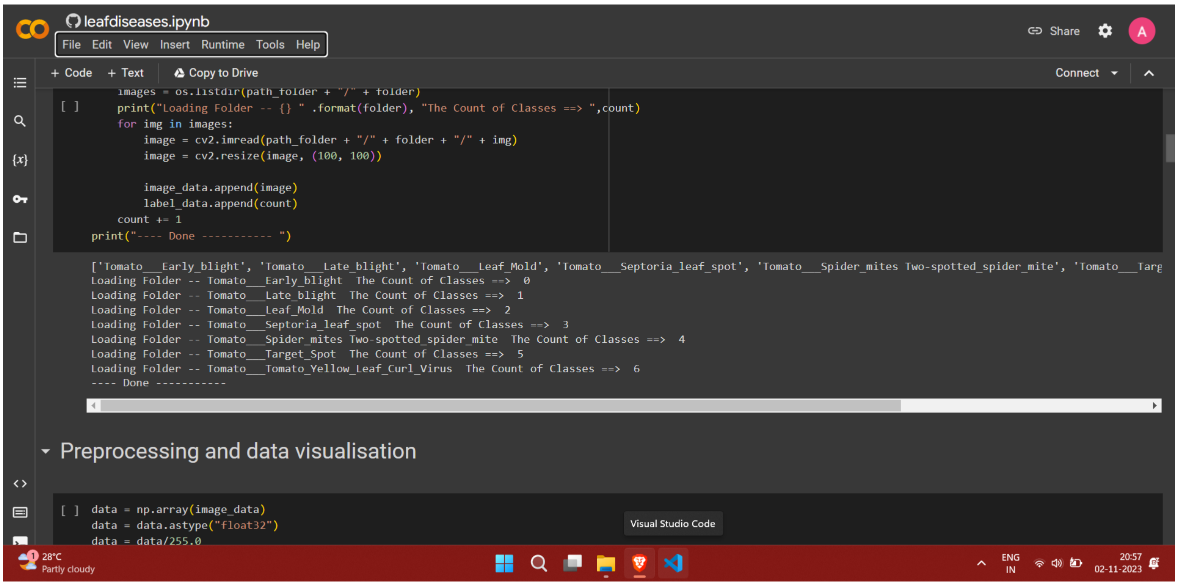


```
#import drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

Reading Data Image

```
path_folder = "/content/drive/My Drive/test"
class_name = os.listdir(path_folder)
class_name.sort()
#class_name = class_name[:4] + class_name[20:23]

class_name = class_name[4:11]
print(class_name)
image_data = []
label_data = []
count = 0
for folder in class_name:
    images = os.listdir(path_folder + "/" + folder)
    print("Loading Folder -- {} " .format(folder), "The Count of Classes ==> ",count)
    for img in images:
        image = cv2.imread(path_folder + "/" + folder + "/" + img)
        image = cv2.resize(image, (100, 100))
```

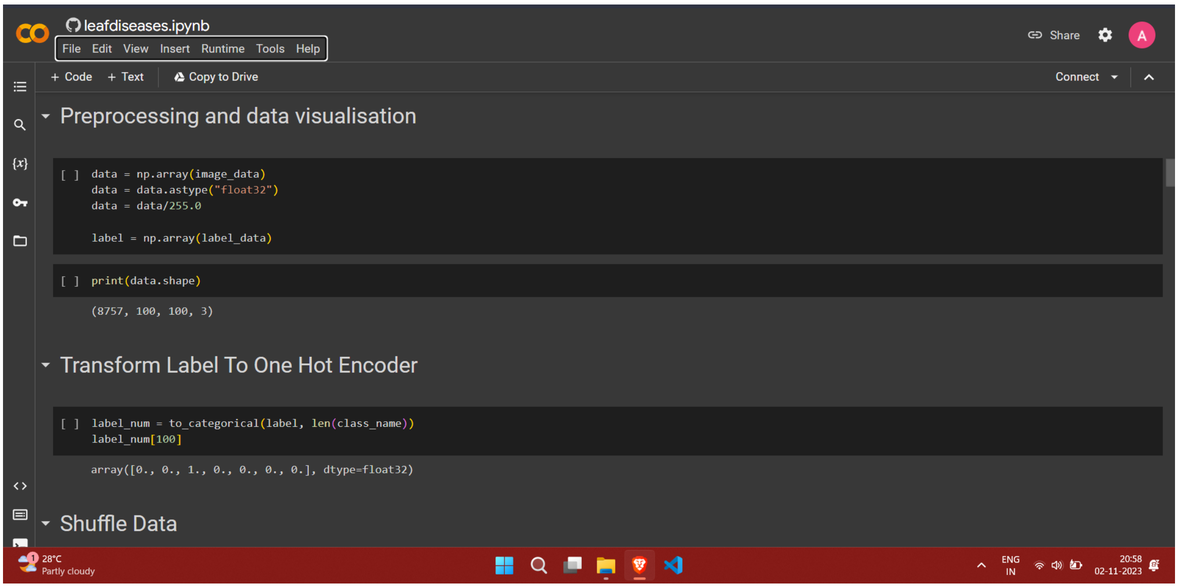


```
    images = os.listdir(path_folder + "/" + folder)
[ ]    print("Loading Folder -- {} " .format(folder), "The Count of Classes ==> ",count)
    for img in images:
        image = cv2.imread(path_folder + "/" + folder + "/" + img)
        image = cv2.resize(image, (100, 100))

        image_data.append(image)
        label_data.append(count)
    count += 1
print("---- Done ----------- ")
```

```
['Tomato___Early_blight', 'Tomato___Late_blight', 'Tomato___Leaf_Mold', 'Tomato___Septoria_leaf_spot', 'Tomato___Spider_mites Two-spotted_spider_mite', 'Tomato___Targ
Loading Folder -- Tomato___Early_blight  The Count of Classes ==>  0
Loading Folder -- Tomato___Late_blight  The Count of Classes ==>  1
Loading Folder -- Tomato___Leaf_Mold  The Count of Classes ==>  2
Loading Folder -- Tomato___Septoria_leaf_spot  The Count of Classes ==>  3
Loading Folder -- Tomato___Spider_mites Two-spotted_spider_mite  The Count of Classes ==>  4
Loading Folder -- Tomato___Target_Spot  The Count of Classes ==>  5
Loading Folder -- Tomato___Tomato_Yellow_Leaf_Curl_Virus  The Count of Classes ==>  6
---- Done -----------
```

Preprocessing and data visualisation

```
data = np.array(image_data)
data = data.astype("float32")
data = data/255.0
```



Preprocessing and data visualisation

```
data = np.array(image_data)
data = data.astype("float32")
data = data/255.0

label = np.array(label_data)
```

```
print(data.shape)
```

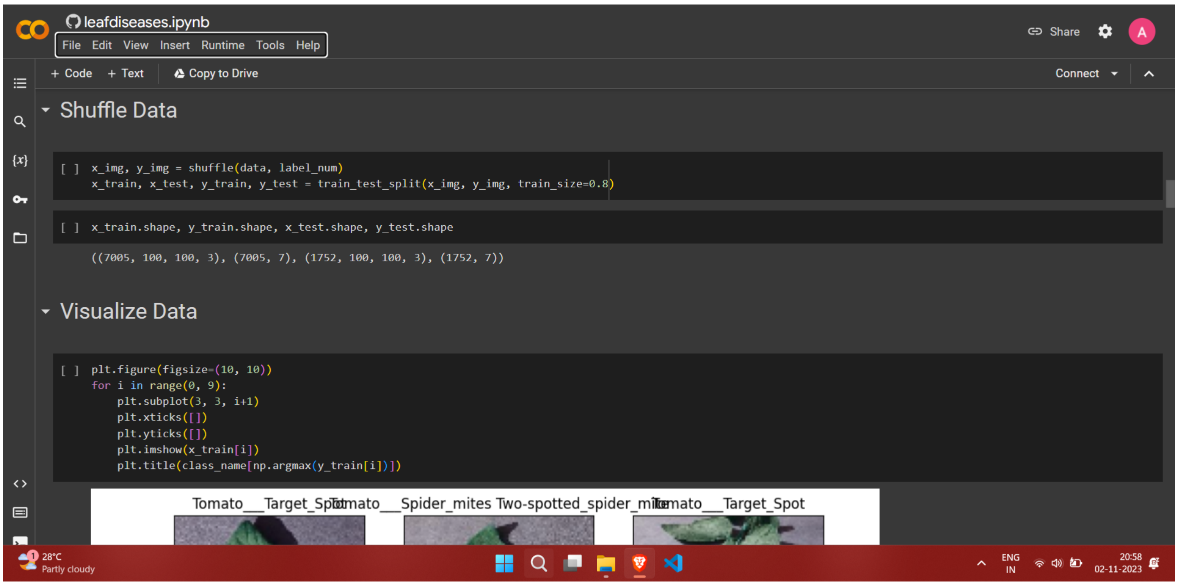
```
(8757, 100, 100, 3)
```

Transform Label To One Hot Encoder

```
label_num = to_categorical(label, len(class_name))
label_num[100]
```

```
array([0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

Shuffle Data



Shuffle Data

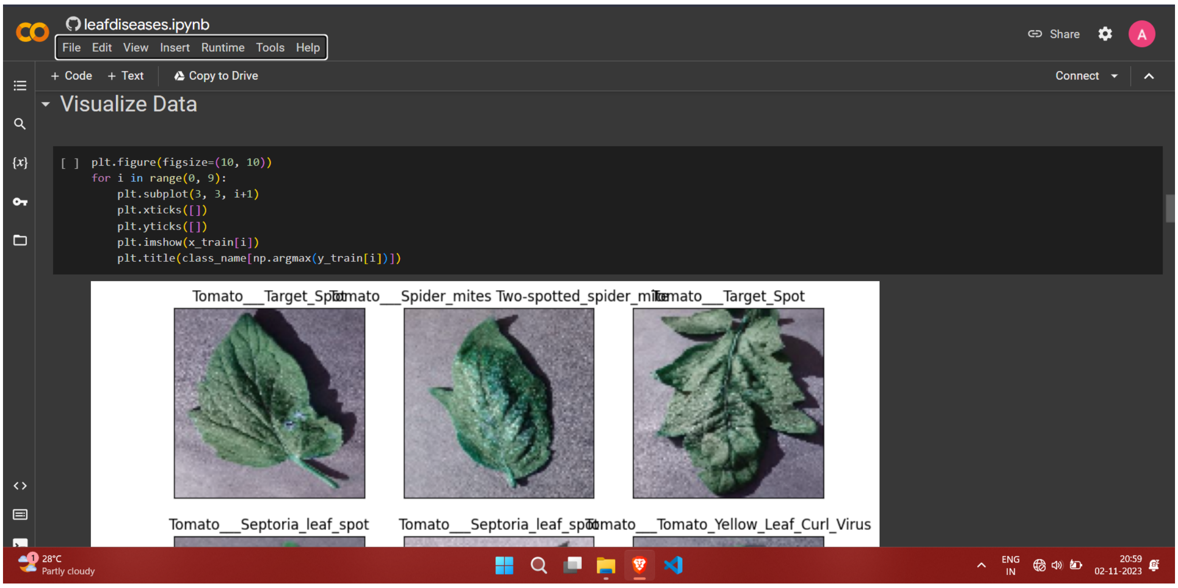
```
[ ] x_img, y_img = shuffle(data, label_num)
    x_train, x_test, y_train, y_test = train_test_split(x_img, y_img, train_size=0.8)
```

```
[ ] x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
((7005, 100, 100, 3), (7005, 7), (1752, 100, 100, 3), (1752, 7))
```

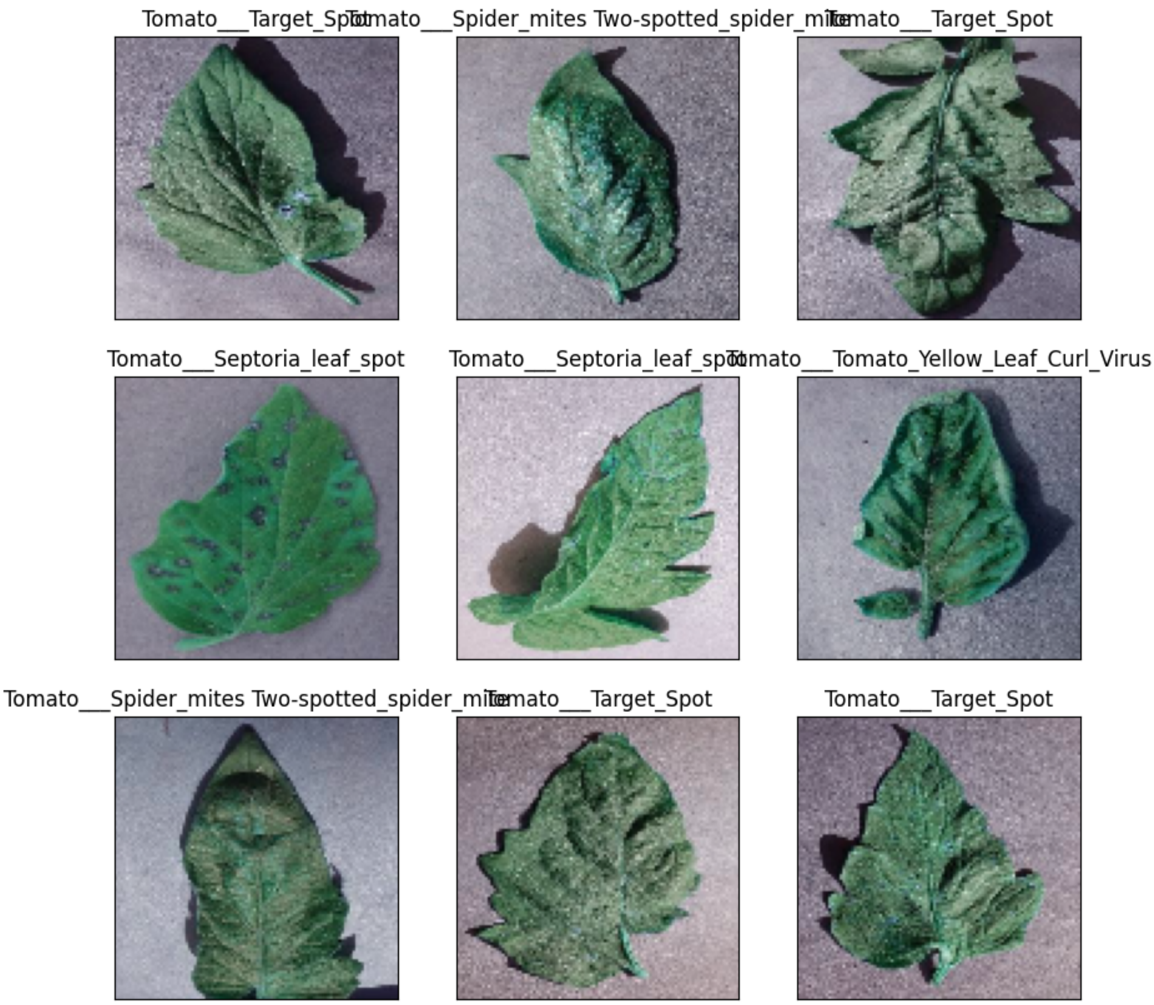
Visualize Data

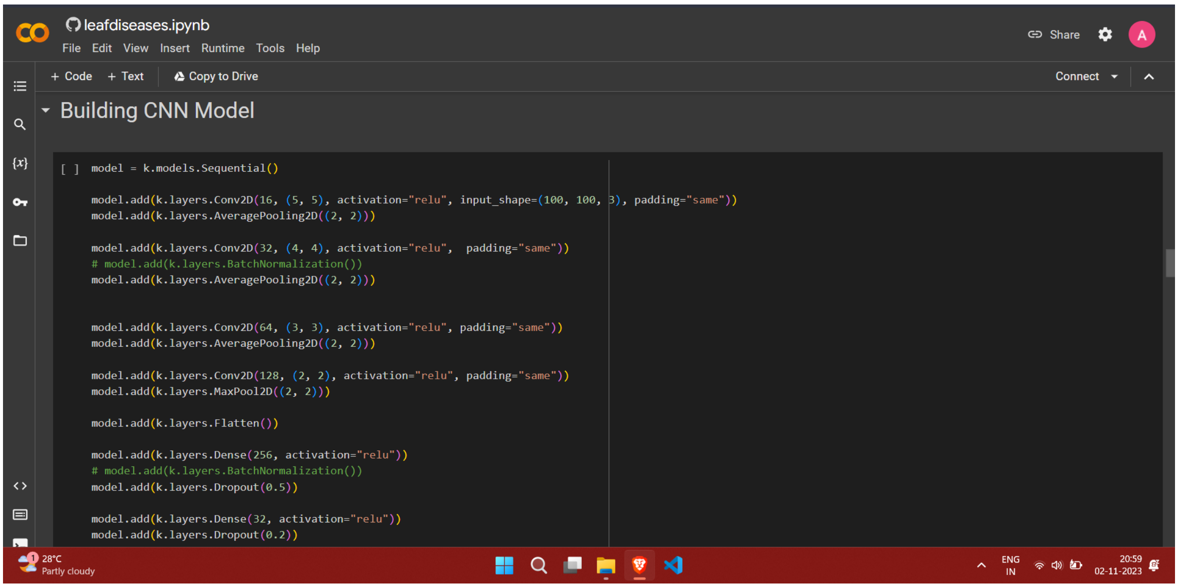
```
[ ] plt.figure(figsize=(10, 10))
    for i in range(0, 9):
        plt.subplot(3, 3, i+1)
        plt.xticks([])
        plt.yticks([])
        plt.imshow(x_train[i])
        plt.title(class_name[np.argmax(y_train[i])])
```



Visualize Data

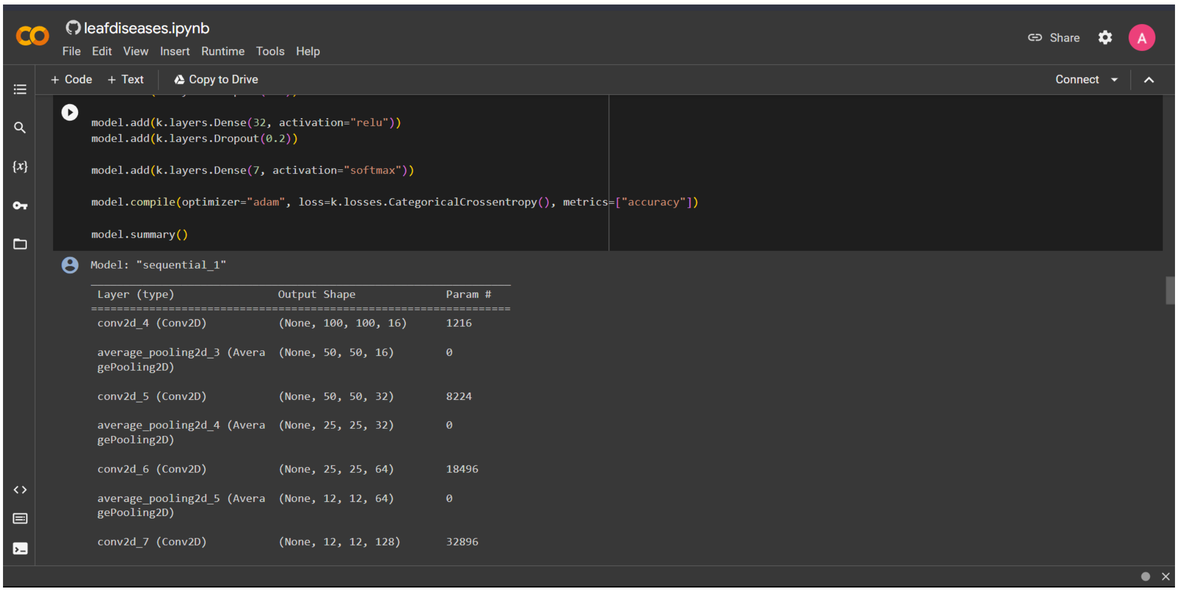
```
[ ] plt.figure(figsize=(10, 10))
    for i in range(0, 9):
        plt.subplot(3, 3, i+1)
        plt.xticks([])
        plt.yticks([])
        plt.imshow(x_train[i])
        plt.title(class_name[np.argmax(y_train[i])])
```





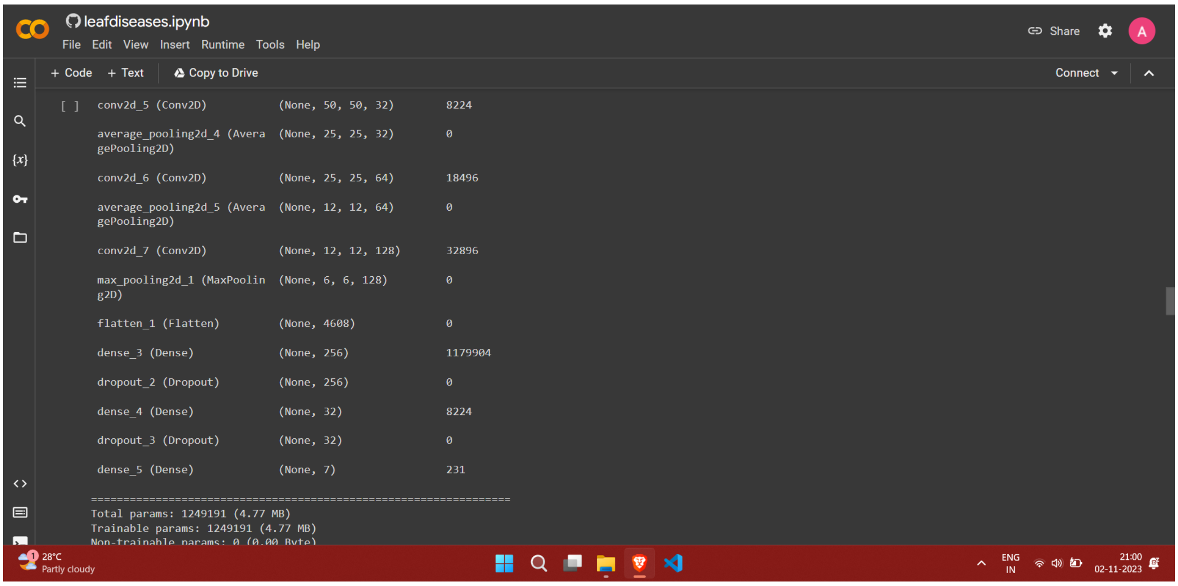
Building CNN Model

```
model = k.models.Sequential()

model.add(k.layers.Conv2D(16, (5, 5), activation="relu", input_shape=(100, 100, 3), padding="same"))
model.add(k.layers.AveragePooling2D((2, 2)))

model.add(k.layers.Conv2D(32, (4, 4), activation="relu",  padding="same"))
# model.add(k.layers.BatchNormalization())
model.add(k.layers.AveragePooling2D((2, 2)))


model.add(k.layers.Conv2D(64, (3, 3), activation="relu", padding="same"))
model.add(k.layers.AveragePooling2D((2, 2)))

model.add(k.layers.Conv2D(128, (2, 2), activation="relu", padding="same"))
model.add(k.layers.MaxPool2D((2, 2)))

model.add(k.layers.Flatten())

model.add(k.layers.Dense(256, activation="relu"))
# model.add(k.layers.BatchNormalization())
model.add(k.layers.Dropout(0.5))

model.add(k.layers.Dense(32, activation="relu"))
model.add(k.layers.Dropout(0.2))
```

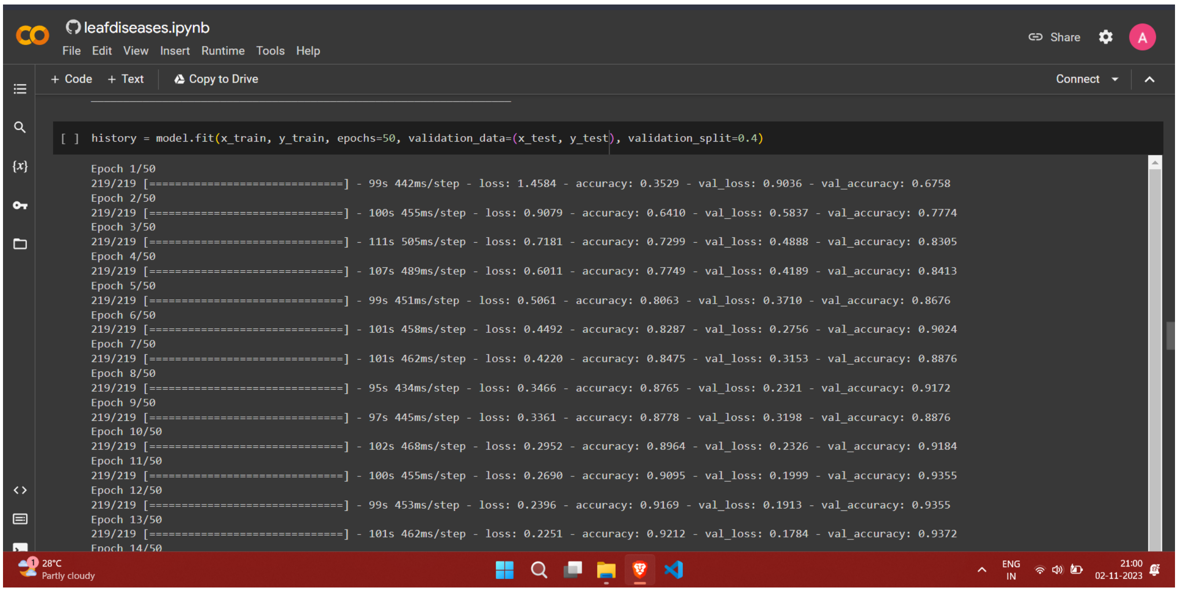


```
model.add(k.layers.Dense(32, activation="relu"))
model.add(k.layers.Dropout(0.2))

model.add(k.layers.Dense(7, activation="softmax"))

model.compile(optimizer="adam", loss=k.losses.CategoricalCrossentropy(), metrics=["accuracy"])

model.summary()
```

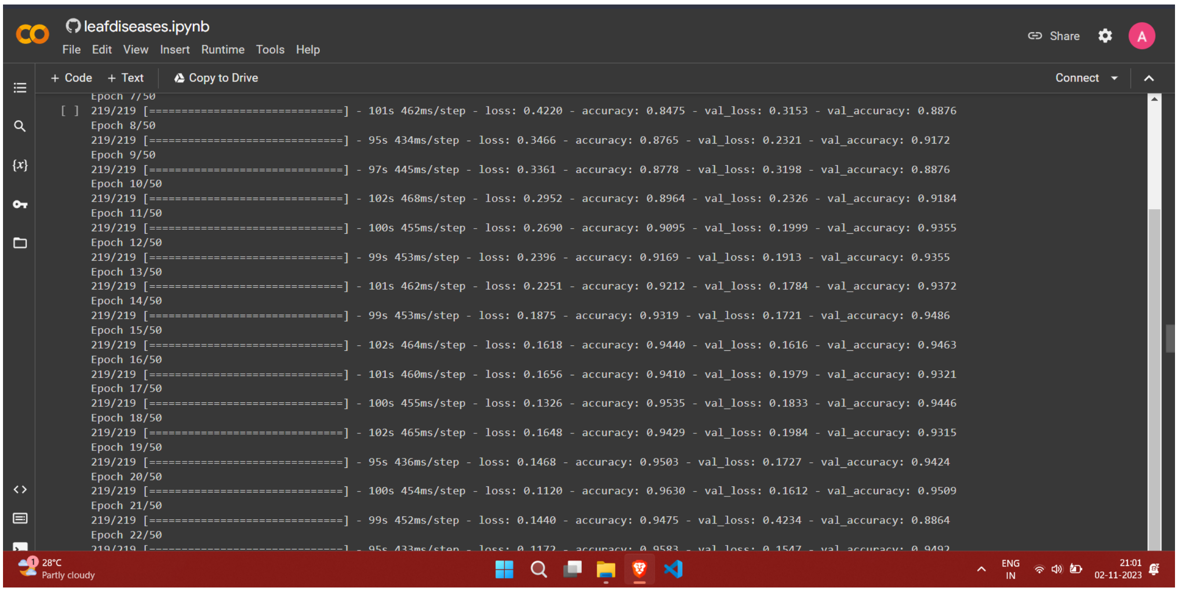
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_4 (Conv2D)           (None, 100, 100, 16)      1216

 average_pooling2d_3 (Avera  (None, 50, 50, 16)        0
 gePooling2D)

 conv2d_5 (Conv2D)           (None, 50, 50, 32)        8224

 average_pooling2d_4 (Avera  (None, 25, 25, 32)        0
 gePooling2D)

 conv2d_6 (Conv2D)           (None, 25, 25, 64)        18496

 average_pooling2d_5 (Avera  (None, 12, 12, 64)        0
 gePooling2D)

 conv2d_7 (Conv2D)           (None, 12, 12, 128)       32896
```



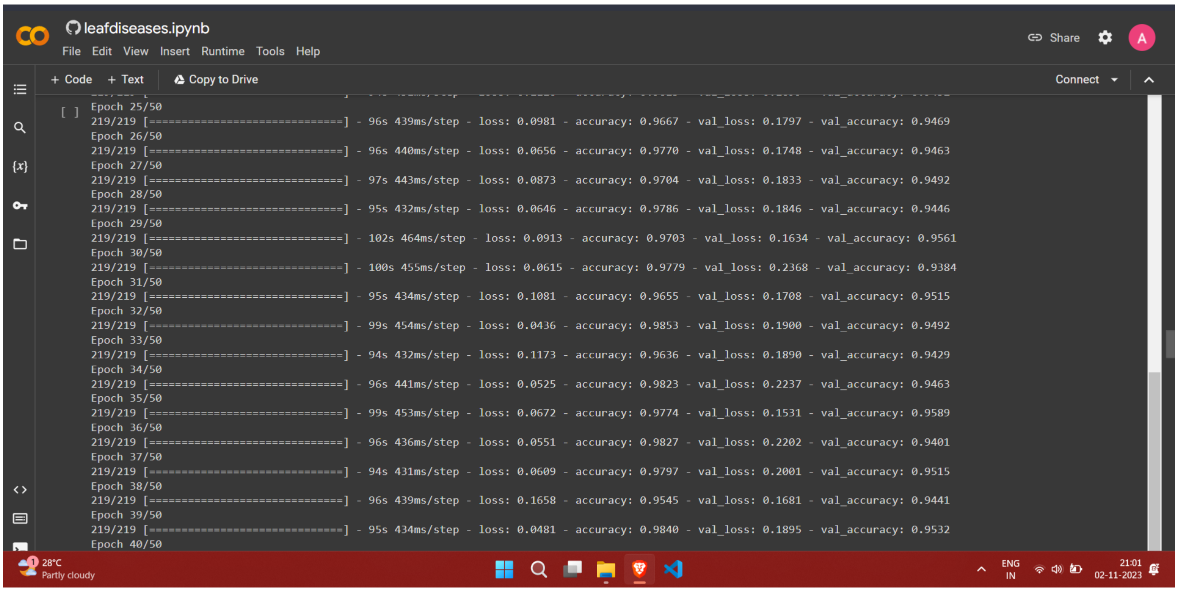
```
[ ] conv2d_5 (Conv2D)           (None, 50, 50, 32)        8224

    average_pooling2d_4 (Avera  (None, 25, 25, 32)        0
    gePooling2D)

    conv2d_6 (Conv2D)           (None, 25, 25, 64)        18496

    average_pooling2d_5 (Avera  (None, 12, 12, 64)        0
    gePooling2D)

    conv2d_7 (Conv2D)           (None, 12, 12, 128)       32896

    max_pooling2d_1 (MaxPoolin  (None, 6, 6, 128)         0
    g2D)

    flatten_1 (Flatten)         (None, 4608)              0

    dense_3 (Dense)             (None, 256)               1179904

    dropout_2 (Dropout)         (None, 256)               0

    dense_4 (Dense)             (None, 32)                8224

    dropout_3 (Dropout)         (None, 32)                0

    dense_5 (Dense)             (None, 7)                 231

   =================================================================
   Total params: 1249191 (4.77 MB)
   Trainable params: 1249191 (4.77 MB)
   Non-trainable params: 0 (0.00 Byte)
```



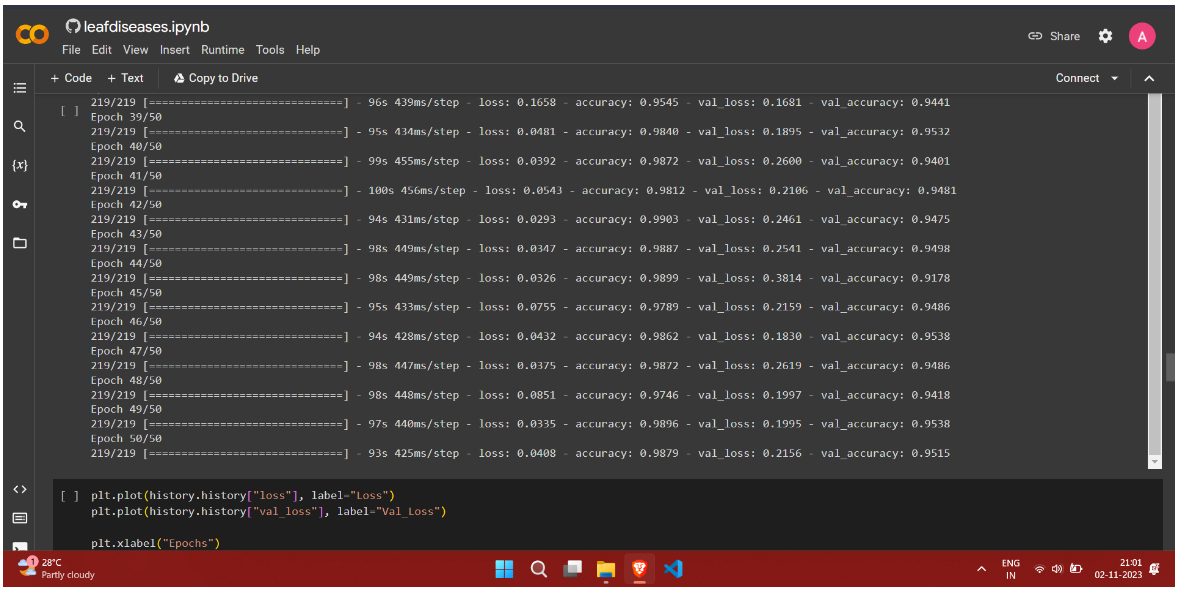
```
[ ] history = model.fit(x_train, y_train, epochs=50, validation_data=(x_test, y_test), validation_split=0.4)

    Epoch 1/50
    219/219 [==============================] - 99s 442ms/step - loss: 1.4584 - accuracy: 0.3529 - val_loss: 0.9036 - val_accuracy: 0.6758
    Epoch 2/50
    219/219 [==============================] - 100s 455ms/step - loss: 0.9079 - accuracy: 0.6410 - val_loss: 0.5837 - val_accuracy: 0.7774
    Epoch 3/50
    219/219 [==============================] - 111s 505ms/step - loss: 0.7181 - accuracy: 0.7299 - val_loss: 0.4888 - val_accuracy: 0.8305
    Epoch 4/50
    219/219 [==============================] - 107s 489ms/step - loss: 0.6011 - accuracy: 0.7749 - val_loss: 0.4189 - val_accuracy: 0.8413
    Epoch 5/50
    219/219 [==============================] - 99s 451ms/step - loss: 0.5061 - accuracy: 0.8063 - val_loss: 0.3710 - val_accuracy: 0.8676
    Epoch 6/50
    219/219 [==============================] - 101s 458ms/step - loss: 0.4492 - accuracy: 0.8287 - val_loss: 0.2756 - val_accuracy: 0.9024
    Epoch 7/50
    219/219 [==============================] - 101s 462ms/step - loss: 0.4220 - accuracy: 0.8475 - val_loss: 0.3153 - val_accuracy: 0.8876
    Epoch 8/50
    219/219 [==============================] - 95s 434ms/step - loss: 0.3466 - accuracy: 0.8765 - val_loss: 0.2321 - val_accuracy: 0.9172
    Epoch 9/50
    219/219 [==============================] - 97s 445ms/step - loss: 0.3361 - accuracy: 0.8778 - val_loss: 0.3198 - val_accuracy: 0.8876
    Epoch 10/50
    219/219 [==============================] - 102s 468ms/step - loss: 0.2952 - accuracy: 0.8964 - val_loss: 0.2326 - val_accuracy: 0.9184
    Epoch 11/50
    219/219 [==============================] - 100s 455ms/step - loss: 0.2690 - accuracy: 0.9095 - val_loss: 0.1999 - val_accuracy: 0.9355
    Epoch 12/50
    219/219 [==============================] - 99s 453ms/step - loss: 0.2396 - accuracy: 0.9169 - val_loss: 0.1913 - val_accuracy: 0.9355
    Epoch 13/50
    219/219 [==============================] - 101s 462ms/step - loss: 0.2251 - accuracy: 0.9212 - val_loss: 0.1784 - val_accuracy: 0.9372
    Epoch 14/50
```



```
    Epoch 7/50
[ ] 219/219 [==============================] - 101s 462ms/step - loss: 0.4220 - accuracy: 0.8475 - val_loss: 0.3153 - val_accuracy: 0.8876
    Epoch 8/50
    219/219 [==============================] - 95s 434ms/step - loss: 0.3466 - accuracy: 0.8765 - val_loss: 0.2321 - val_accuracy: 0.9172
    Epoch 9/50
    219/219 [==============================] - 97s 445ms/step - loss: 0.3361 - accuracy: 0.8778 - val_loss: 0.3198 - val_accuracy: 0.8876
    Epoch 10/50
    219/219 [==============================] - 102s 468ms/step - loss: 0.2952 - accuracy: 0.8964 - val_loss: 0.2326 - val_accuracy: 0.9184
    Epoch 11/50
    219/219 [==============================] - 100s 455ms/step - loss: 0.2690 - accuracy: 0.9095 - val_loss: 0.1999 - val_accuracy: 0.9355
    Epoch 12/50
    219/219 [==============================] - 99s 453ms/step - loss: 0.2396 - accuracy: 0.9169 - val_loss: 0.1913 - val_accuracy: 0.9355
    Epoch 13/50
    219/219 [==============================] - 101s 462ms/step - loss: 0.2251 - accuracy: 0.9212 - val_loss: 0.1784 - val_accuracy: 0.9372
    Epoch 14/50
    219/219 [==============================] - 99s 453ms/step - loss: 0.1875 - accuracy: 0.9319 - val_loss: 0.1721 - val_accuracy: 0.9486
    Epoch 15/50
    219/219 [==============================] - 102s 464ms/step - loss: 0.1618 - accuracy: 0.9440 - val_loss: 0.1616 - val_accuracy: 0.9463
    Epoch 16/50
    219/219 [==============================] - 101s 460ms/step - loss: 0.1656 - accuracy: 0.9410 - val_loss: 0.1979 - val_accuracy: 0.9321
    Epoch 17/50
    219/219 [==============================] - 100s 455ms/step - loss: 0.1326 - accuracy: 0.9535 - val_loss: 0.1833 - val_accuracy: 0.9446
    Epoch 18/50
    219/219 [==============================] - 102s 465ms/step - loss: 0.1648 - accuracy: 0.9429 - val_loss: 0.1984 - val_accuracy: 0.9315
    Epoch 19/50
    219/219 [==============================] - 95s 436ms/step - loss: 0.1468 - accuracy: 0.9503 - val_loss: 0.1727 - val_accuracy: 0.9424
    Epoch 20/50
    219/219 [==============================] - 100s 454ms/step - loss: 0.1120 - accuracy: 0.9630 - val_loss: 0.1612 - val_accuracy: 0.9509
    Epoch 21/50
    219/219 [==============================] - 99s 452ms/step - loss: 0.1440 - accuracy: 0.9475 - val_loss: 0.4234 - val_accuracy: 0.8864
    Epoch 22/50
    219/219 [==============================] - 95s 433ms/step - loss: 0.1172 - accuracy: 0.9583 - val_loss: 0.1547 - val_accuracy: 0.9492
```

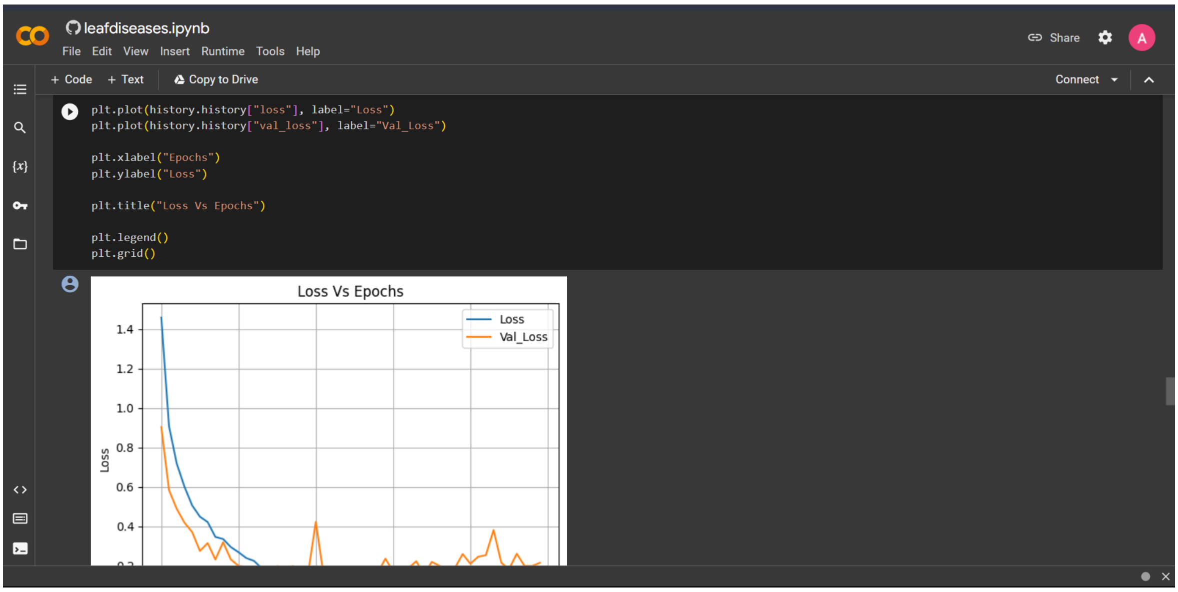


```
[ ] Epoch 25/50
    219/219 [==============================] - 96s 439ms/step - loss: 0.0981 - accuracy: 0.9667 - val_loss: 0.1797 - val_accuracy: 0.9469
    Epoch 26/50
    219/219 [==============================] - 96s 440ms/step - loss: 0.0656 - accuracy: 0.9770 - val_loss: 0.1748 - val_accuracy: 0.9463
    Epoch 27/50
    219/219 [==============================] - 97s 443ms/step - loss: 0.0873 - accuracy: 0.9704 - val_loss: 0.1833 - val_accuracy: 0.9492
    Epoch 28/50
    219/219 [==============================] - 95s 432ms/step - loss: 0.0646 - accuracy: 0.9786 - val_loss: 0.1846 - val_accuracy: 0.9446
    Epoch 29/50
    219/219 [==============================] - 102s 464ms/step - loss: 0.0913 - accuracy: 0.9703 - val_loss: 0.1634 - val_accuracy: 0.9561
    Epoch 30/50
    219/219 [==============================] - 100s 455ms/step - loss: 0.0615 - accuracy: 0.9779 - val_loss: 0.2368 - val_accuracy: 0.9384
    Epoch 31/50
    219/219 [==============================] - 95s 434ms/step - loss: 0.1081 - accuracy: 0.9655 - val_loss: 0.1708 - val_accuracy: 0.9515
    Epoch 32/50
    219/219 [==============================] - 99s 454ms/step - loss: 0.0436 - accuracy: 0.9853 - val_loss: 0.1900 - val_accuracy: 0.9492
    Epoch 33/50
    219/219 [==============================] - 94s 432ms/step - loss: 0.1173 - accuracy: 0.9636 - val_loss: 0.1890 - val_accuracy: 0.9429
    Epoch 34/50
    219/219 [==============================] - 96s 441ms/step - loss: 0.0525 - accuracy: 0.9823 - val_loss: 0.2237 - val_accuracy: 0.9463
    Epoch 35/50
    219/219 [==============================] - 99s 453ms/step - loss: 0.0672 - accuracy: 0.9774 - val_loss: 0.1531 - val_accuracy: 0.9589
    Epoch 36/50
    219/219 [==============================] - 96s 436ms/step - loss: 0.0551 - accuracy: 0.9827 - val_loss: 0.2202 - val_accuracy: 0.9401
    Epoch 37/50
    219/219 [==============================] - 94s 431ms/step - loss: 0.0609 - accuracy: 0.9797 - val_loss: 0.2001 - val_accuracy: 0.9515
    Epoch 38/50
    219/219 [==============================] - 96s 439ms/step - loss: 0.1658 - accuracy: 0.9545 - val_loss: 0.1681 - val_accuracy: 0.9441
    Epoch 39/50
    219/219 [==============================] - 95s 434ms/step - loss: 0.0481 - accuracy: 0.9840 - val_loss: 0.1895 - val_accuracy: 0.9532
    Epoch 40/50
```

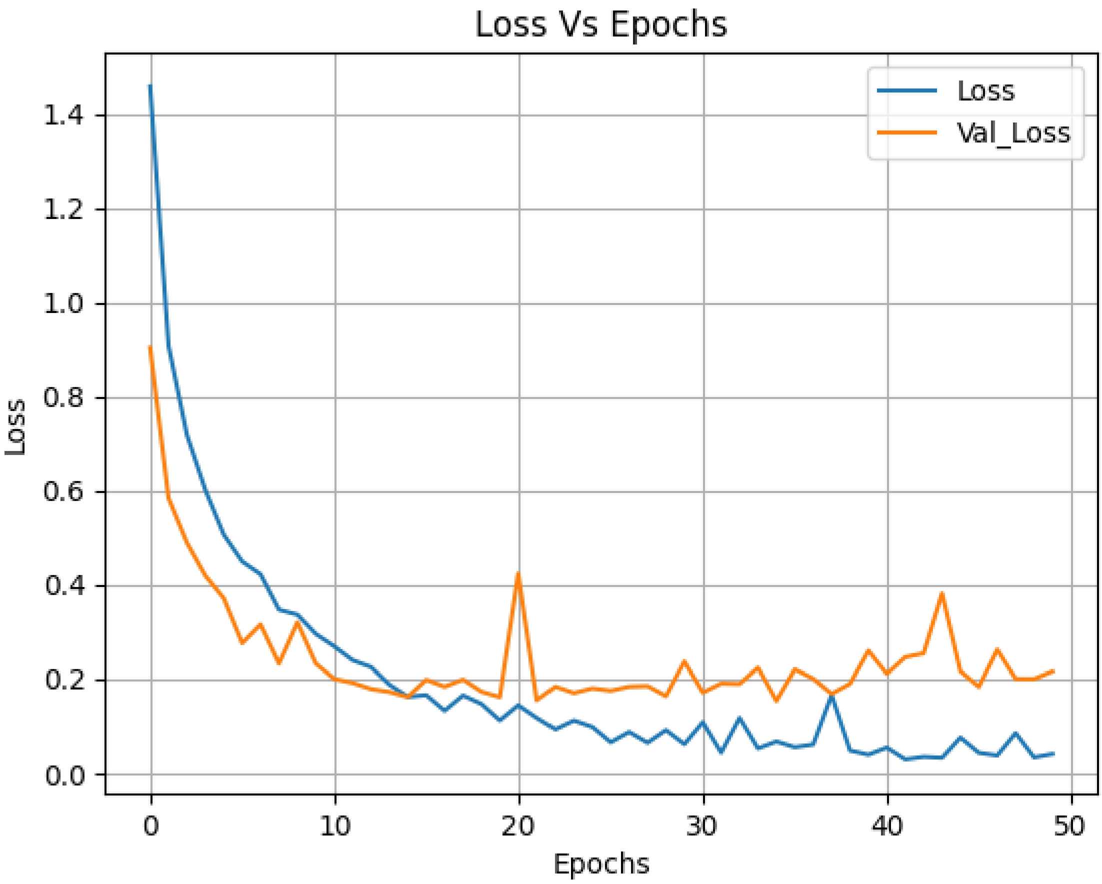


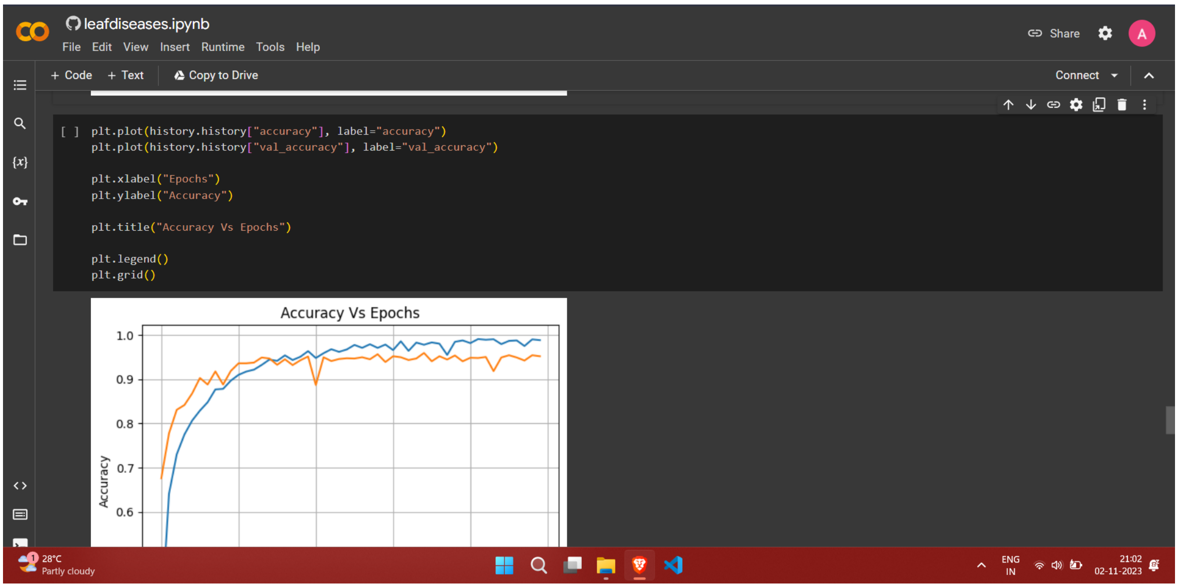
```
    219/219 [==============================] - 96s 439ms/step - loss: 0.1658 - accuracy: 0.9545 - val_loss: 0.1681 - val_accuracy: 0.9441
[ ] Epoch 39/50
    219/219 [==============================] - 95s 434ms/step - loss: 0.0481 - accuracy: 0.9840 - val_loss: 0.1895 - val_accuracy: 0.9532
    Epoch 40/50
    219/219 [==============================] - 99s 455ms/step - loss: 0.0392 - accuracy: 0.9872 - val_loss: 0.2600 - val_accuracy: 0.9401
    Epoch 41/50
    219/219 [==============================] - 100s 456ms/step - loss: 0.0543 - accuracy: 0.9812 - val_loss: 0.2106 - val_accuracy: 0.9481
    Epoch 42/50
    219/219 [==============================] - 94s 431ms/step - loss: 0.0293 - accuracy: 0.9903 - val_loss: 0.2461 - val_accuracy: 0.9475
    Epoch 43/50
    219/219 [==============================] - 98s 449ms/step - loss: 0.0347 - accuracy: 0.9887 - val_loss: 0.2541 - val_accuracy: 0.9498
    Epoch 44/50
    219/219 [==============================] - 98s 449ms/step - loss: 0.0326 - accuracy: 0.9899 - val_loss: 0.3814 - val_accuracy: 0.9178
    Epoch 45/50
    219/219 [==============================] - 95s 433ms/step - loss: 0.0755 - accuracy: 0.9789 - val_loss: 0.2159 - val_accuracy: 0.9486
    Epoch 46/50
    219/219 [==============================] - 94s 428ms/step - loss: 0.0432 - accuracy: 0.9862 - val_loss: 0.1830 - val_accuracy: 0.9538
    Epoch 47/50
    219/219 [==============================] - 98s 447ms/step - loss: 0.0375 - accuracy: 0.9872 - val_loss: 0.2619 - val_accuracy: 0.9486
    Epoch 48/50
    219/219 [==============================] - 98s 448ms/step - loss: 0.0851 - accuracy: 0.9746 - val_loss: 0.1997 - val_accuracy: 0.9418
    Epoch 49/50
    219/219 [==============================] - 97s 440ms/step - loss: 0.0335 - accuracy: 0.9896 - val_loss: 0.1995 - val_accuracy: 0.9538
    Epoch 50/50
    219/219 [==============================] - 93s 425ms/step - loss: 0.0408 - accuracy: 0.9879 - val_loss: 0.2156 - val_accuracy: 0.9515
```

```
[ ] plt.plot(history.history["loss"], label="Loss")
    plt.plot(history.history["val_loss"], label="Val_Loss")

    plt.xlabel("Epochs")
```

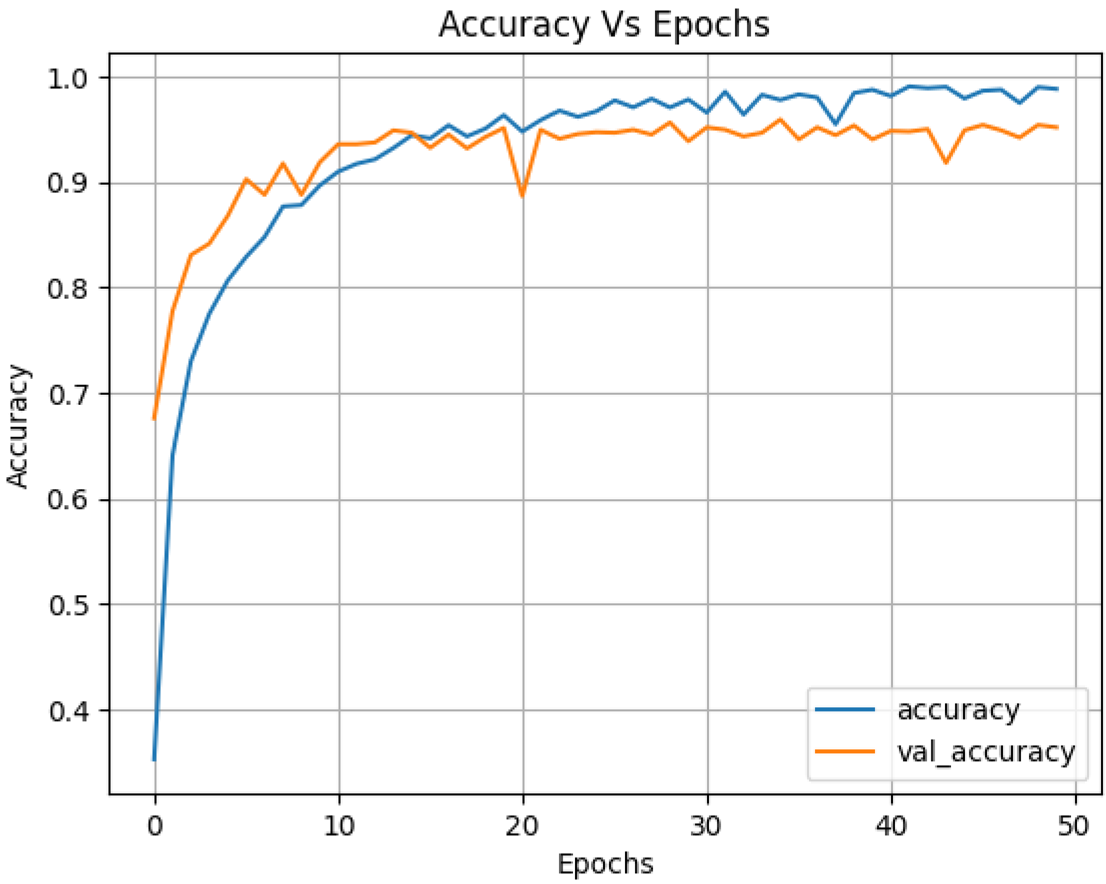


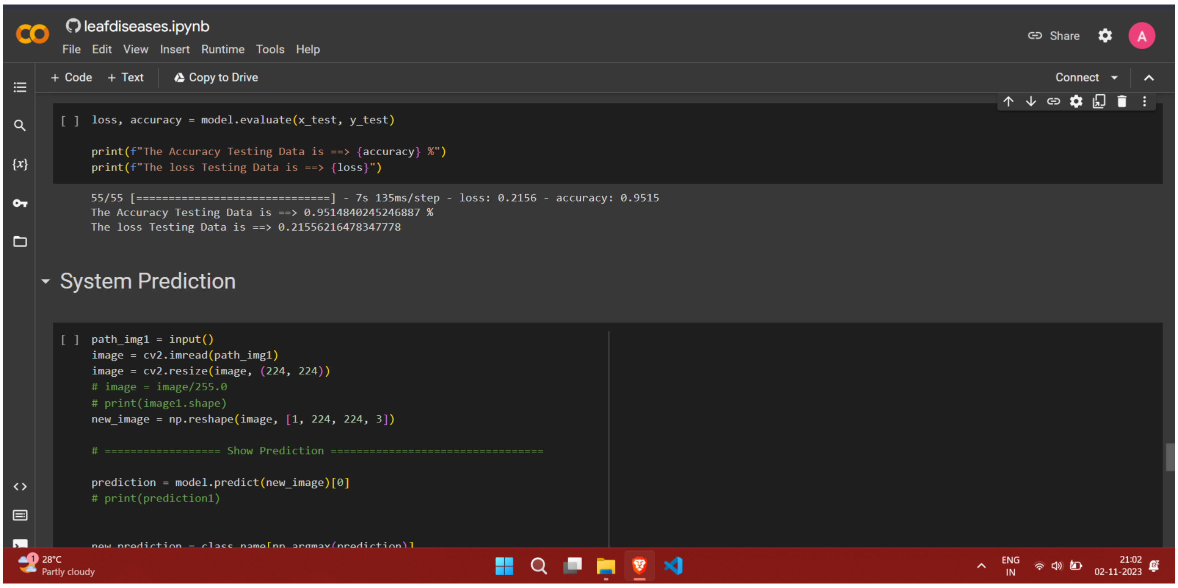
```
plt.plot(history.history["loss"], label="Loss")
plt.plot(history.history["val_loss"], label="Val_Loss")

plt.xlabel("Epochs")
plt.ylabel("Loss")

plt.title("Loss Vs Epochs")

plt.legend()
plt.grid()
```





```
plt.plot(history.history["accuracy"], label="accuracy")
plt.plot(history.history["val_accuracy"], label="val_accuracy")

plt.xlabel("Epochs")
plt.ylabel("Accuracy")

plt.title("Accuracy Vs Epochs")

plt.legend()
plt.grid()
```

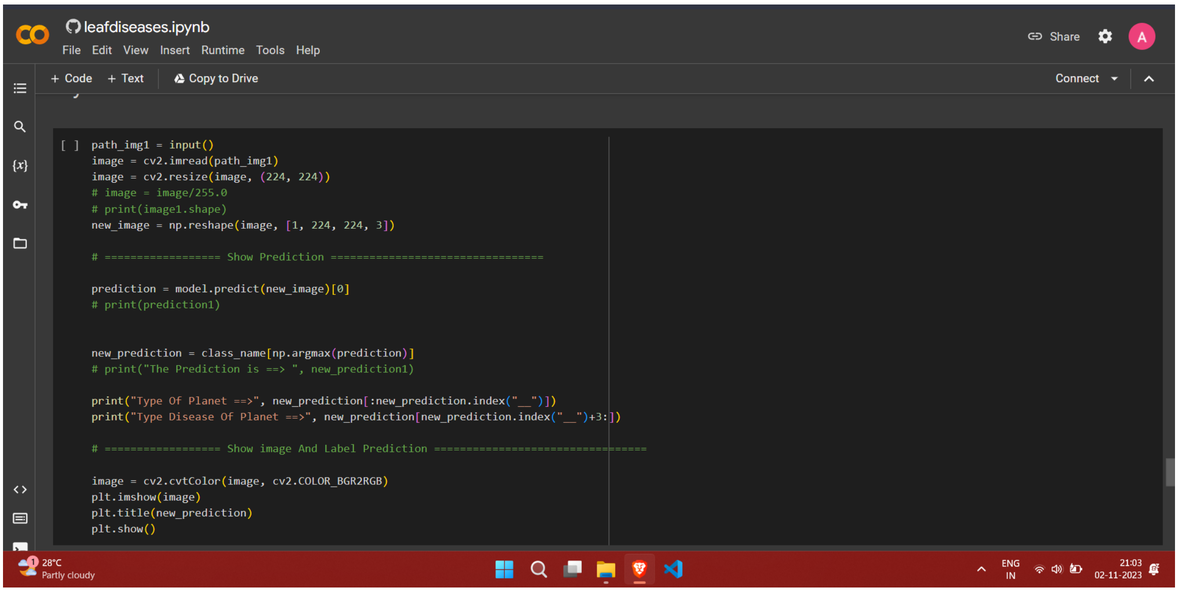




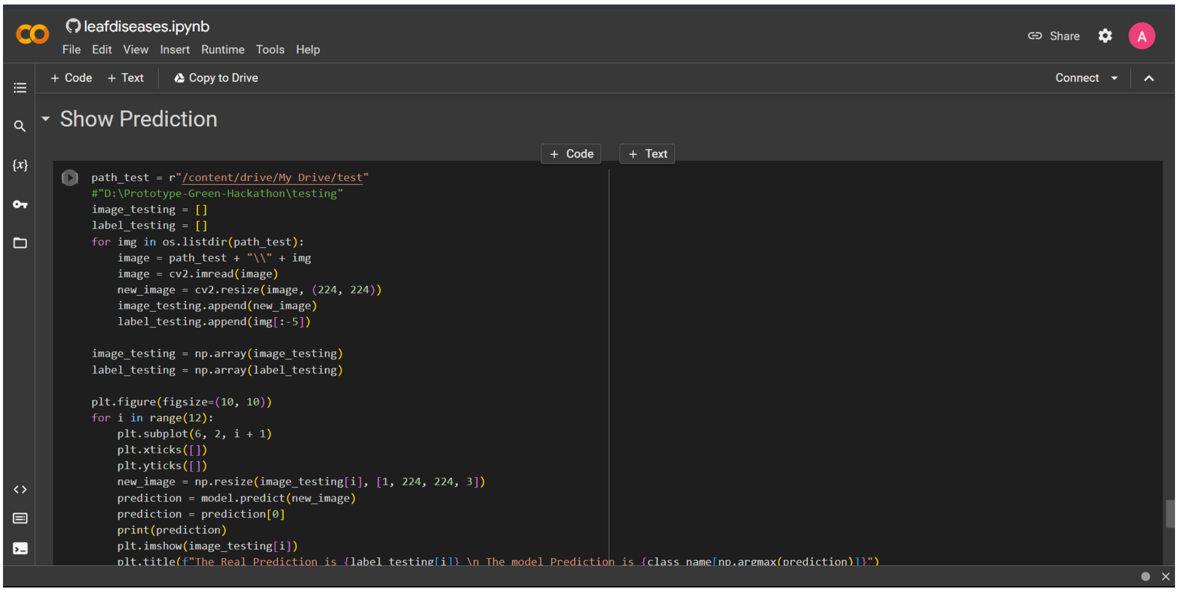
```
[ ] loss, accuracy = model.evaluate(x_test, y_test)

    print(f"The Accuracy Testing Data is ==> {accuracy} %")
    print(f"The loss Testing Data is ==> {loss}")

    55/55 [==============================] - 7s 135ms/step - loss: 0.2156 - accuracy: 0.9515
    The Accuracy Testing Data is ==> 0.9514840245246887 %
    The loss Testing Data is ==> 0.21556216478347778
```

System Prediction

```
[ ] path_img1 = input()
    image = cv2.imread(path_img1)
    image = cv2.resize(image, (224, 224))
    # image = image/255.0
    # print(image1.shape)
    new_image = np.reshape(image, [1, 224, 224, 3])

    # ================= Show Prediction ================================

    prediction = model.predict(new_image)[0]
    # print(prediction1)


    new_prediction = class_name[np.argmax(prediction)]
```

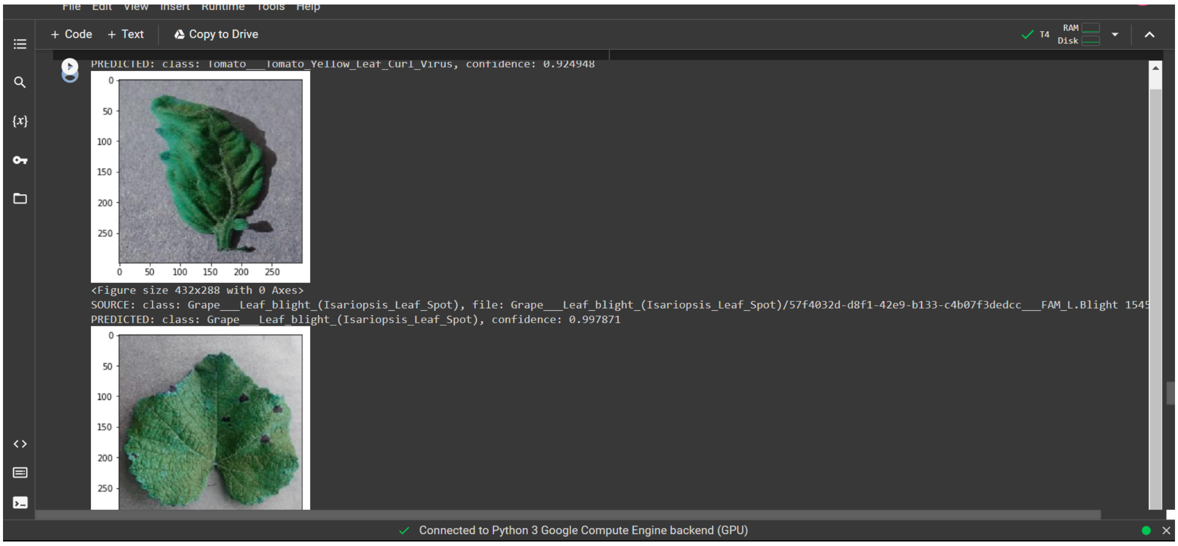


```
[ ] path_img1 = input()
    image = cv2.imread(path_img1)
    image = cv2.resize(image, (224, 224))
    # image = image/255.0
    # print(image1.shape)
    new_image = np.reshape(image, [1, 224, 224, 3])

    # ================= Show Prediction ================================

    prediction = model.predict(new_image)[0]
    # print(prediction1)


    new_prediction = class_name[np.argmax(prediction)]
    # print("The Prediction is ==> ", new_prediction1)

    print("Type Of Planet ==>", new_prediction[:new_prediction.index("__")])
    print("Type Disease Of Planet ==>", new_prediction[new_prediction.index("__")+3:])

    # ================= Show image And Label Prediction ================================

    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    plt.imshow(image)
    plt.title(new_prediction)
    plt.show()
```



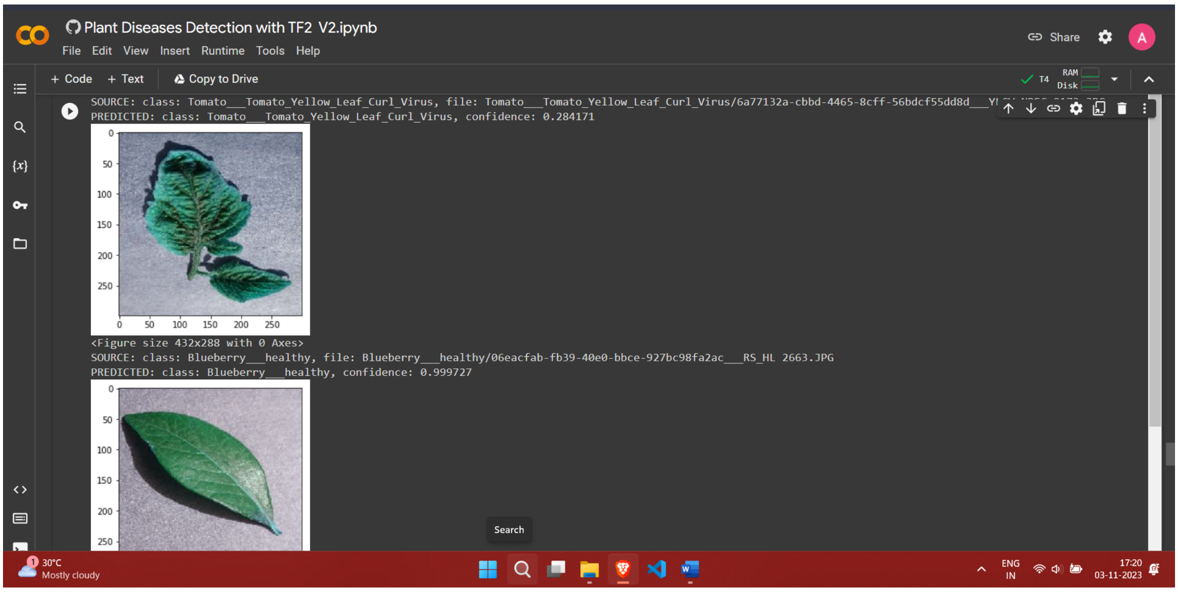
Show Prediction

```
path_test = r"/content/drive/My Drive/test"
#"D:\Prototype-Green-Hackathon\testing"
image_testing = []
label_testing = []
for img in os.listdir(path_test):
    image = path_test + "\\" + img
    image = cv2.imread(image)
    new_image = cv2.resize(image, (224, 224))
    image_testing.append(new_image)
    label_testing.append(img[:-5])

image_testing = np.array(image_testing)
label_testing = np.array(label_testing)

plt.figure(figsize=(10, 10))
for i in range(12):
    plt.subplot(6, 2, i + 1)
    plt.xticks([])
    plt.yticks([])
    new_image = np.resize(image_testing[i], [1, 224, 224, 3])
    prediction = model.predict(new_image)
    prediction = prediction[0]
    print(prediction)
    plt.imshow(image_testing[i])
    plt.title(f"The Real Prediction is {label_testing[i]} \n The model Prediction is {class_name[np.argmax(prediction)]}")
```



```
PREDICTED: class: Tomato___Tomato_Yellow_Leaf_Curl_Virus, confidence: 0.924948
<Figure size 432x288 with 0 Axes>
SOURCE: class: Grape___Leaf_blight_(Isariopsis_Leaf_Spot), file: Grape___Leaf_blight_(Isariopsis_Leaf_Spot)/57f4032d-d8f1-42e9-b133-c4b07f3dedcc___FAM_L.Blight 1545
PREDICTED: class: Grape___Leaf_blight_(Isariopsis_Leaf_Spot), confidence: 0.997871
```

Connected to Python 3 Google Compute Engine backend (GPU)



```
SOURCE: class: Tomato___Tomato_Yellow_Leaf_Curl_Virus, file: Tomato___Tomato_Yellow_Leaf_Curl_Virus/6a77132a-cbbd-4465-8cff-56bdcf55dd8d___Y
PREDICTED: class: Tomato___Tomato_Yellow_Leaf_Curl_Virus, confidence: 0.284171
```

```
<Figure size 432x288 with 0 Axes>
SOURCE: class: Blueberry___healthy, file: Blueberry___healthy/06eacfab-fb39-40e0-bbce-927bc98fa2ac___RS_HL 2663.JPG
PREDICTED: class: Blueberry___healthy, confidence: 0.999727
```