**PLANT DISEASE DETECTION**

(**CNN)**

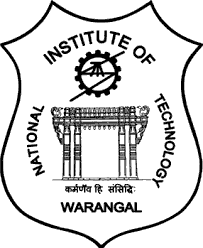
**Internship Project Report**

**BY**

**N. SIDDHARTHA @SR UNIVERSITY**

**K. SHIVA SHANKAR @** **SR UNIVERSITY**

**S. RAHUL SUNNY @** **SR UNIVERSITY**



**UNDER THE GUIDANCE OF**

**Prof L. Anjaneyulu**

**PROFESSOR**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY, WARANGAL

NATIONAL INSTITUTE OF TECHNOLOGY

WARANGAL - 506 004

**CERTIFICATE**

This is to certify that N. SIDDHARTHA, K.SHIVA SHANKAR, S. RAHUL SUNNY

of SR University, Hasanparthy have successfully completed a Project

titled “plant disease detection ”, as part of Summer Internship Program under my

guidance at National Institute of Technology, Warangal,

Telangana, during 10-07-2024 to 31-07-2024.

**Prof L. Anjaneyulu, ECE**

**Visvesvaraya Centre for Skill Development**

**NIT, Warangal**

**TABLEOFCONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.no** | **Content** | **Pageno** |
| 1 | Abstract | 05 |
| 3 | Introduction | 07 |
| 8 | Proposed Solution | 11 |
| 9 | System Architecture | 12 |
| 10 | Flow chart | 14 |
| 11 | Data flow diagram | 15 |
| 12 | Simulation Setup | 16 |
| 13 | Implementation | 18 |
| 14 | Result and Analysis | 21 |
| 15 | Learning Outcome | 23 |
| 16 | Conclusion with Challenges | 25 |
| 17 | Future Scope | 26 |
| 18 | References | 27 |

**List of figures**

|  |  |  |
| --- | --- | --- |
| **Fig no** | **Figure Title** | **Pageno** |
| 1 | Use-Case Diagram |  |
| 2 | System architecture |  |
| 3 | Flowchart |  |
| 4 | Data flow Diagram |  |
| 5 | Dataset Gathering |  |
| 6 | Pre-processing of video |  |
| 7 | Overview of model |  |
| 8 | Training and Validation loss |  |
| 9 | Training and validation Accuracy |  |
| 10 | Homepage |  |
| 11 | Uploading Video |  |

**List of Acronyms**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Acronym** | **Abbreviation** |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |

## ABSTRACT

Plants and crops that are infected by pests have an impact on the country’s agricultural production. Usually, farmers or professionals keep a close eye on the plants in order to discover and identify diseases. However, this procedure is frequently time-consuming, costly, and imprecise. Plant disease detection can be done by looking for a spot on the diseased leaves. The goal of this document is to create a disease recognition model that is supported by leaf image classification. To detect plant disease, we are utilizing image processing with a convolution neural network (CNN). A convolution neural network is a form of artificial neural network that is specifically intended to process pixel input and is used in image recognition. **As** Plant disease is an ongoing challenge for smallholder farmers, which threatens income and food security. The recent revolution in smartphone penetration and computer vision models has created an opportunity for image classification in agriculture. Convolutional Neural Networks (CNNs) are considered state-of-the-art in image recognition and offer the ability to provide a prompt and definite diagnosis. In this paper, the performance of a pre-trained ResNet34 model in detecting crop disease is investigated. The developed model is deployed as a web application and is capable of recognizing 7 plant diseases out of healthy leaf tissue. A dataset containing 8,685 leaf images; captured in a controlled environment, is established for training and validating the model. Validation results show that the proposed method can achieve an accuracy of 97.2% and an F1 score of greater than 96.5%. This demonstrates the technical feasibility of CNNs in classifying plant diseases and presents a path towards AI solutions for small holder farmers.

# 

# 

# INTRODUCTION

Agricultural production is a very old means of obtaining food. It is a vital source of income for people all around the world. No one can exist in our world without food. Plants are crucial not only for humans, but also for animals who rely on them for food, oxygen, and other necessities. The government and experts are taking significant initiatives to enhance food production, and they are working successfully in the real world. When a plant becomes afflicted with a disease, all living organisms in the environment are affected in some way. This plant disease can affect anywhere on the plant, including the stem, leaf, and branch. Even the types of illnesses that impact plants, such as bacterial and fungal diseases .etc can differ. The illness that impacts the crops will be determined by factors such as climate. There are a large number of people that are food insecure. This occurs as a result of insufficient food crop output. Even significant climate changes will have an impact on plant development. This type of natural tragedy is unavoidable. Early detection of plant disease aids in the prevention of large-scale crop losses. Farmers must apply the appropriate insecticides for their crops. Too many pesticides are harmful to crops and farmland. Getting expert advice will help you avoid misusing chemicals on plants. Plants have been the focus of many researchers to aid farmers and others involved in agriculture. When a disease is visible to the naked eye, it is straightforward to detect. The illness may be discovered and treated early if the farmer has sufficient information and monitors the crops on a regular basis. However, this phase only exists when the disease is extreme or crop output is low. Then there are the different innovations. Farmers will benefit from the introduction of automated disease detection tools. This approach yields outcomes that are suitable for both little and large-scale agricultural cultivation. Importantly, the results are precise, and the disorders are detected in a very short amount of time. These technologies rely heavily on deep learning and neural networks to function. Deep Convolutional Neural Network is utilized in this study to identify infected and healthy leaves, as well as to detect illness in afflicted plants. The CNN model is designed to suit both healthy and sick leaves; photos are used to train the model, and the output is determined by the input leaf.

# PROBLEM STATEMENT

As the world population grows, the demand for food production also increases. However, one of the significant challenges faced by farmers and agricultural experts is the spread of plant diseases. Plant diseases can lead to significant yield losses and affect food security.

The agricultural sector faces a significant challenge in combating plant diseases, which can lead to substantial crop losses, affecting food security and economic stability. Early detection and intervention are crucial for mitigating these impacts.

# REQUIREMENT ANALYSIS

### FunctionalRequirements:

* + Real-timedeepfakedetectioncapabilities.
  + User-friendlyinterfaceforeaseof use.
  + Supportforcontinuousupdatestoimprovedetection accuracy.

### Non-FunctionalRequirements:

* + Performance:Thesystemshouldprovidereal-time analysisofvideo content.
  + Scalability:Thesystem should handlealargevolumeofvideo data.
  + Reliability:Thesystemshouldoperatewithhighavailability.

### DataRequirements:

* + TrainingData:Diverseandrepresentativedatasetsofbothgenuineandsynthetic videos.
  + DataFormats:Supportforcommonvideoformats(e.g.,MP4).
  + DataHandling: Protocolsforupdating andmanagingthetrainingdataset.

### RequiredTechnologies:

1. **ProgrammingLanguages:**
   1. Pythonforitsversatilityandextensivelibrary support.

### Libraries:

* 1. TensorFlowandKerasforimplementingandtrainingneuralnetworkmodels.
  2. OpenCVforvideoprocessingandframe extraction.
  3. NumPyfornumerical operations.
  4. Matplotliborothervisualizationlibrariesforresultrepresentation.

### Platform:

* 1. GoogleColabforGPU-acceleratedtraining.

### GPUAcceleration:

* 1. UtilizeGPUresourcesonplatformslikeGoogleColabforfastermodeltraining and inference.

1. **Technical Risks:**

# RISK ANALYSIS

* 1. Compatibilityissuesbetweenlibrariesandplatforms.
  2. Challengesinoptimizingthemodel forGPUacceleration.

### DataQualityandAvailability:

* 1. Risksassociatedwithacquiringdiverseandrepresentativetrainingdata.
  2. Ensuringthecontinuousavailabilityandqualityofdataforongoingmodel updates.

### ModelAccuracyandFalsePositives/Negatives:

* 1. Risksrelatedtotheinherentuncertaintyindeepfakedetection.
  2. Thepotentialchallengeofminimizingfalsepositivesandfalsenegativesduring real-world deployment.

### FeasibilityAnalysis:

1. **TechnicalFeasibility:**
   1. AvailabilityofexpertiseinPython,TensorFlow,andGPUacceleration.
   2. AccesstoGPUresources onplatformslikeGoogleColab.

### Operational Feasibility:

* 1. Assessthepracticalityofintegratingthesystemintoexistingworkflows.
  2. Evaluateuseracceptance andtrainingrequirements.

### ScheduleFeasibility:

* 1. Evaluatethetimerequired fordevelopment, testing,and deployment.
  2. Identifypotentialfactorsthatmayimpacttheprojecttimeline,suchasdata availability and model optimization challenges.

# PROPOSED SOLUTION

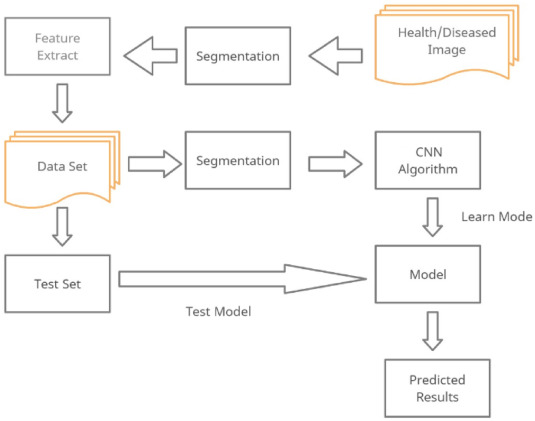
The primary objective is to build a Convolutional Neural Network (CNN) model using TensorFlow/Keras that can effectively classify plant leaves as healthy or diseased. The model should be trained to achieve high accuracy and robustness in detecting various diseases affecting plants. Additionally, the project involves data preprocessing steps such as image resizing, normalization, and augmentation to enhance model performance. Upon successful model development, the project will evaluate the model's performance using metrics like accuracy, precision, recall, and F1-score. A confusion matrix will be generated to gain insights into the model's predictions and identify any potential areas for improvement. The ultimate goal is to deploy the trained model into a user-friendly interface (web or mobile app) where users can upload images of plant leaves for real-time disease detection. The success criteria include achieving a high accuracy rate (>90%) in classifying healthy and diseased plant images and delivering a reliable and intuitive interface for users to access the disease detection.

The project aims to develop a machine learning model for accurately detecting diseases in plant leaves based on images. With the global demand for food production rising, early detection of plant diseases is crucial to prevent yield losses and ensure food security. The dataset consists of images of plant leaves categorized into healthy and diseased classes, with labels indicating the specific diseases.

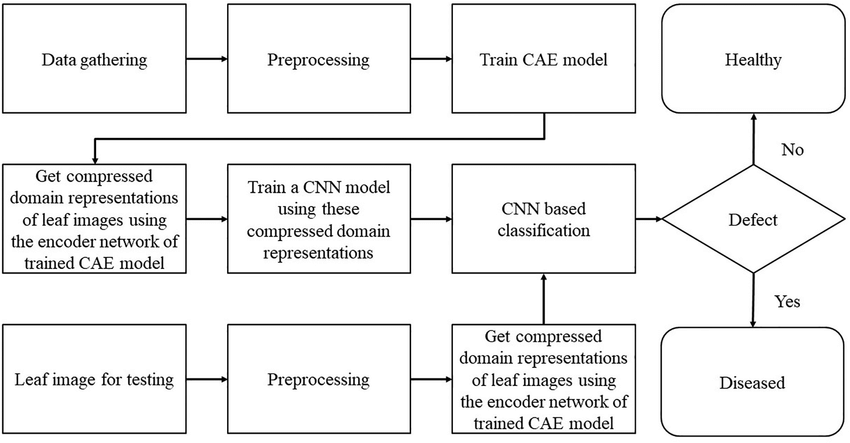
# SYSTEM ARCHITECTURE

1. **Data Collection and Preprocessing:**
   * **Data Sources:** Collect a diverse dataset of plant images, including healthy plants and various diseased conditions, with annotations indicating disease types. This dataset can be sourced from agricultural research institutions, online repositories, or data collection efforts.
   * **Data Preprocessing:** Preprocess the images by standardizing sizes, enhancing contrast, and augmenting the dataset. Techniques such as resizing, normalization, cropping, and data augmentation can be applied to improve model robustness and generalization.
2. **Model Development:**
   * **Convolutional Neural Network (CNN):** Build a deep learning model using a Convolutional Neural Network architecture. CNNs are well-suited for image classification tasks and can automatically learn relevant features from input images.
   * **Transfer Learning:** Utilize pre-trained CNN models (e.g., VGG16, ResNet, Inception) as a base and fine-tune them using the collected dataset. Transfer learning allows leveraging knowledge from large datasets to improve performance on specific tasks with limited data.
   * **Model Optimization:** Optimize the model architecture, hyperparameters, and loss functions to enhance accuracy, minimize overfitting, and improve training efficiency.
3. **Training and Validation:**
   * **Data Splitting:** Split the annotated dataset into training, validation, and test sets. The training set is used to train the model, while the validation set helps tune hyperparameters and prevent overfitting. The test set evaluates the model's performance on unseen data.
   * **Model Evaluation:** Evaluate the trained model using metrics such as accuracy, precision, recall, F1-score, and confusion matrix to assess its performance in disease classification and identification.
   * **Training Pipeline:** Implement a training pipeline that feeds batches of preprocessed images into the CNN model. Use techniques like batch normalization, dropout, and early stopping to improve training stability and convergence.
4. **Deployment and Integration:**
   * Develop a user-friendly web application for plant disease detection.
   * Deploy the trained model on a cloud server for processing.
   * Implement an API endpoint for image uploads and result retrieval.
5. **Feedback and Iteration:**
   * Gather feedback from farmers and experts on model accuracy and usability.
   * Use feedback to update the model and improve disease detection capabilities.
   * Continuously monitor and analyze data trends to adapt to evolving disease patterns.

# FLOWCHART



# DATA FLOW DIAGRAM



# SIMULATION SETUP

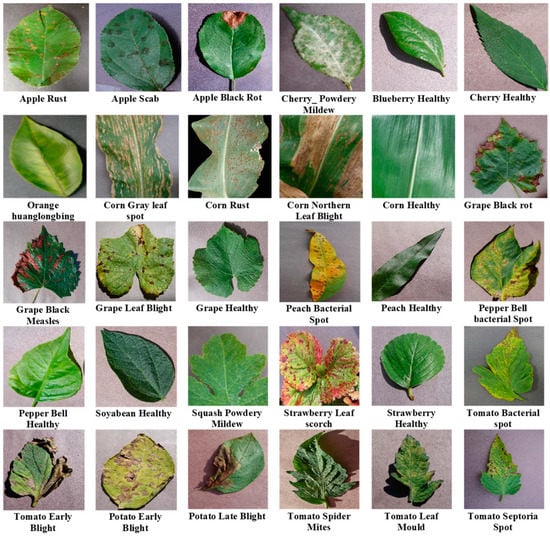
### Dateset:

1. **Data quality assurance:** To keep our dataset's integrity, quality control procedures were put in place. Make sure all the data that was used in training was high in quality and relevance, which include removing of any duplicate or subpar samples.
2. **DataDiversity:**Ourdatasetincludedawidevarietyofsubjects,backgrounds,andlighting conditions to accurately reflect a wide range of real-world scenarios. For the model to effectively generalize to a wide range of situations and content types, diversity is crucial.
3. **Maintenance:**Wearededicatedtomaintainingthequalityofthedatalongafterithasbeen prepared. We removed the corrupted videos after preprocessing. We are aware of how crucial regular upkeep and updates are to the model's continued effectiveness in adapting to new deep fake challenges and techniques.

### Toolsused:

1. **Platform Choice:** Due to the collaborative online environment and GPU resources provided by Google Collab, which greatly sped up our workflow, we chose this platform as the main one for the implementation of our project.
2. **Tools for Collaboration:** Google Collab was a meaningful change for promoting communication, version control, and sharing among the project team, streamlining the collaborative aspects of our endeavour.
3. **Programming Language of Choice:** Python, which is renowned for its adaptability and extensive libraries, served as our main language of choice for model development and training.
4. **Deep Learning Frameworks:** PyTorch was essential in the implementation of Long Short-Term Memory (LSTM) based Recurrent Neural Networks (RNNs) for modelling temporal dynamics in video content, while TensorFlow was crucial in the design and training of Convolutional Neural Networks (CNNs) for spatial feature extraction.
5. **Data Analysis Libraries**: We tapped into the power of Pandas to perform robust data manipulationandanalysis.Improveourabilitytovisualiseandanalysedata,wealsoused Matplotlib and Seaborn to produce visually appealing and instructive graphs and plots.

# IMPLEMENTATION

1. **DATAGATHERING**: The firststepintacklinganymachinelearningtaskisacquiringthe necessary data. This data can be gathered from publicly available sources such as Kaggle ormeticulouslycraftedtocreateacustomizeddataset,whichwasourchosenapproachfor our project. Our method involved combining various datasets from external sources, including both genuine and manipulated videos collected from FaceForensics++,Kaggle Deepfake Detection Challenge, and Celeb Deepfakes. Additionally, we created a comprehensive global CSV file containing labels for every video in the datasets we obtained.Weundertookthisprocessofmergingdifferentdatasetstoimprovetheprecision of our project.
2. **Preprocessing:**
3. 

### TRAINING AND MODELLING:

The following steps were part of the training process:

**Data Preparation:** This step involves cleaning and preprocessing the raw data to make it suitable for training the machine learning model. It includes tasks such as resizing images, normalizing pixel values, augmenting the dataset to increase diversity, and encoding labels for categorical data.

**Model Architecture:** the design and structure of the machine learning model. This includes determining the number of layers, types of layers (e.g., convolutional, pooling, dense), activation functions, and other architectural decisions that define how the model processes input data and generates output predictions

**Feeding the Dataset:** For model training, we fed the model our pre-processed dataset. Evaluate the model's performance during training and avoid overfitting, the dataset was split into training and validation sets.

**Training and Optimization:** Involves training the model on the training set using optimization algorithms such as stochastic gradient descent (SGD), Adam, or RMSprop. The goal is to minimize the model's loss function by adjusting weights and biases iteratively. Optimization techniques such as learning rate scheduling, regularization (e.g., dropout), and early stopping may be applied to improve training efficiency and prevent overfitting.

**Evaluation:** During training, the model's performance is evaluated on the validation set at regular intervals (epochs). Evaluation metrics such as accuracy, precision, recall, F1-score, and loss are calculated to assess how well the model is learning and generalizing from the training data.

**Model Selection:** Depending on the requirements of convolutional neural network (CNN), we chose the model that performed the best on our evaluation metrics, which may have included accuracy, precision, recall, or even F1-score

**Testing:** Following training and model selection, we evaluated the model on a separate test dataset to judge how well it generalized and performed on new data.

### PREDICTION:

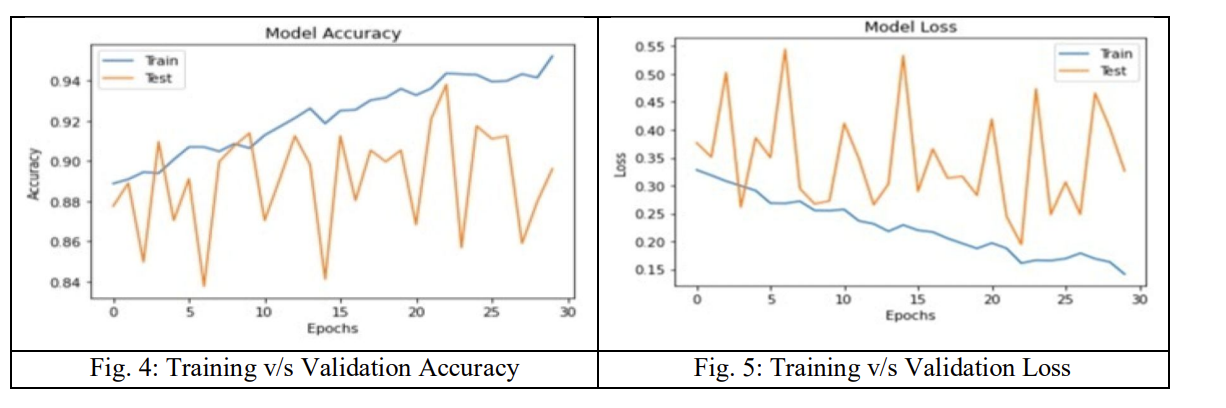
In a plant disease detection project using Convolutional Neural Networks (CNNs), the predictions provided by the model typically revolve around identifying and diagnosing diseases or abnormalities in plants based on visual symptoms captured in images. Here's an overview of the predictions the project can offer:

1. **Healthy vs. Diseased Plants:** The primary prediction involves distinguishing between healthy plants and those affected by diseases or pests. The CNN model can classify input images into two categories: "healthy" or "diseased," providing insights into the overall health status of plants.
2. **Specific Disease Identification:** For multi-class classification tasks, the CNN can predict specific diseases or disorders affecting plants. This includes identifying common plant diseases such as powdery mildew, leaf spot, rust, blight, or nutrient deficiencies based on visual symptoms exhibited in the images.
3. **Severity Assessment:** In addition to disease identification, the model may provide predictions related to the severity or extent of the detected disease. For instance, it can categorize diseases into mild, moderate, or severe based on the intensity and spread of symptoms observed in the images.
4. **Recommendations and Actions:** Based on the predictions, the system can generate recommendations or actions for plant management. This may include suggesting specific treatments, interventions, or agricultural practices to mitigate the impact of diseases, prevent further spread, and promote plant health.
5. **Detection of Anomalies or Abnormalities:** Beyond disease detection, the CNN can also predict anomalies or abnormalities that do not fit typical disease patterns. This could include identifying unusual patterns, growth irregularities, or environmental stressors that affect plant health but may not be linked to specific diseases.
6. **Quantitative Insights:** The model may provide quantitative insights such as disease prevalence rates, distribution maps of diseases across regions or crops, and trends over time. These insights can inform decision-making for farmers, agronomists, or policymakers regarding crop management strategies and resource allocation.

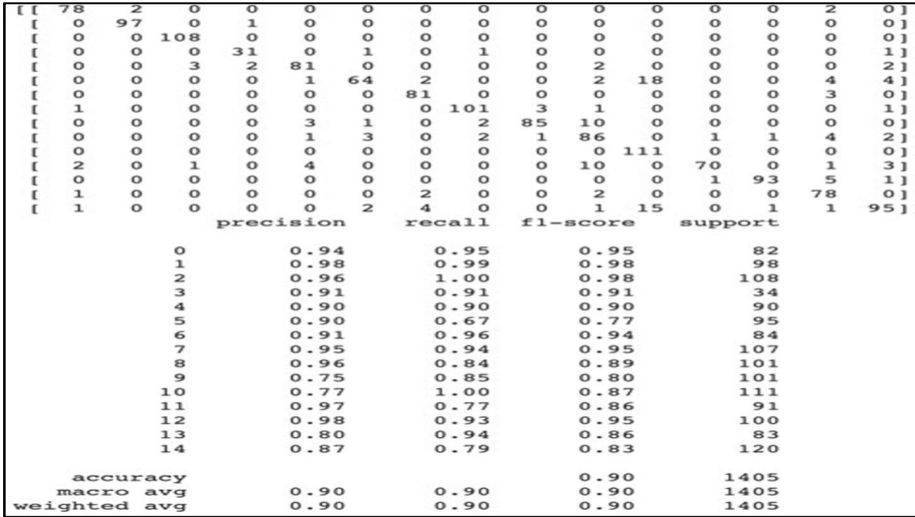
# RESULT COMPARISON AND ANALYSIS

After generating predictions using a plant disease detection model, it's crucial to perform a comprehensive result comparison and analysis to evaluate the model's performance and gain insights into its strengths and weaknesses. This analysis begins by gathering ground truth data, consisting of actual labels (healthy or diseased) for the plant images used in predictions. Constructing a confusion matrix provides a detailed breakdown of the model's predictions against the ground truth labels, including true positives, true negatives, false positives, and false negatives. From this matrix, metrics such as accuracy, precision, recall (sensitivity), specificity, and F1-score are calculated to assess classification accuracy and the model's ability to correctly identify positive and negative instances. High accuracy suggests effective classification, while balanced precision and recall indicate robustness in minimizing false positives and false negatives. Deeper analysis includes error examination to identify misclassifications and patterns for model refinement. Highlighting model strengths, such as high accuracy in certain disease classifications, and areas for improvement, like misclassifying specific diseases, guides recommendations for model enhancement, such as diverse training data collection or hyperparameter fine-tuning. Proposing future work, such as incorporating additional features or deploying the model in real-time monitoring, ensures continuous improvement and application effectiveness in plant disease detection systems.

Training vs validation



Confusion matrix



**SOURCE CODE**

from google.colab import drive

drive.mount('/content/drive')

import numpy as np

import matplotlib.pyplot as plt

import tensorflow as tf

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout from google.colab import files

import zipfile

import io zip\_data = io.BytesIO(uploaded['archive (10).zip'])

with zipfile.ZipFile(zip\_data, 'r') as zip\_ref:

zip\_ref.extractall('/content/drive/MyDrive')

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

train\_datagen = ImageDataGenerator(rescale=1./255, shear\_range=0.2, zoom\_range=0.2, horizontal\_flip=True)

test\_datagen = ImageDataGenerator(rescale=1./255)

train\_generator = train\_datagen.flow\_from\_directory('/content/drive/MyDrive', target\_size=(64, 64), batch\_size=32, class\_mode='categorical')

test\_generator = test\_datagen.flow\_from\_directory('/content/drive/MyDrive', target\_size=(64, 64), batch\_size=32, class\_mode='categorical')

# CONVOLUTIONAL NEURAL NETWORK

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 3)),

MaxPooling2D(pool\_size=(2, 2)),

Conv2D(64, (3, 3), activation='relu'),

MaxPooling2D(pool\_size=(2, 2)),

Flatten(),

Dense(128, activation='relu'),

Dropout(0.5),

Dense(6, activation='softmax') # Assuming you have 6 classes

])

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(train\_generator, epochs=1, validation\_data=test\_generator)

import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='accuracy')

plt.plot(history.history['val\_accuracy'], label = 'val\_accuracy')

plt.ylabel('Accuracy')

plt.ylim([0, 1])

plt.legend(loc='lower right')

plt.show()

from sklearn.metrics import confusion\_matrix

import numpy as np

predictions = model.predict(test\_generator)

true\_labels = test\_generator.classes

binary\_predictions = np.round(predictions).flatten()

# Calculate the confusion matrix

cm = confusion\_matrix(true\_labels, binary\_predictions)

print("Confusion Matrix:")

print(cm)

# Visualize the confusion matrix (optional)

import seaborn as sns

import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.xlabel('Predicted Labels')

plt.ylabel('True Labels')

plt.title('Confusion Matrix')

plt.show()

train\_accuracy = history.history['accuracy']

val\_accuracy = history.history['val\_accuracy']

train\_loss = history.history['loss']

val\_loss = history.history['val\_loss']

epochs = range(1, len(train\_accuracy) + 1)

import matplotlib.pyplot as plt

# Plot training vs validation accuracy

plt.plot(epochs, train\_accuracy, 'b', label='Training Accuracy')

plt.plot(epochs, val\_accuracy, 'r', label='Validation Accuracy')

plt.title('Training and Validation Accuracy')

plt.xlabel('Epochs')

plt.ylabel('Accuracy')

plt.legend()

plt.show()

# Plot training vs validation loss

plt.plot(epochs, train\_loss, 'b', label='Training Loss')

plt.plot(epochs, val\_loss, 'r', label='Validation Loss')

plt.title('Training and Validation Loss')

plt.xlabel('Epochs')

plt.ylabel('Loss')

plt.legend()

plt.show()

test\_loss, test\_accuracy = model.evaluate(test\_generator)

print(f'Test Loss: {test\_loss}')

print(f'Test Accuracy: {test\_accuracy}')

# LEARNING OUTCOME

1. **Technical Proficiency:** Participants develop proficiency in machine learning techniques such as data preprocessing, model training, evaluation, and result analysis. They gain hands-on experience in working with image data, implementing neural network architectures, and optimizing models for accuracy and generalization.
2. **Problem-Solving Skills:** Through addressing challenges like class imbalance, model interpretability, and feature importance, learners hone their problem-solving skills. They learn to apply appropriate techniques to overcome common issues encountered in machine learning projects, fostering a deeper understanding of algorithmic solutions.
3. **Data Interpretation:** Participants enhance their ability to interpret and analyze data, especially in the context of agricultural or biological datasets. They learn to extract meaningful insights from complex data structures, identify patterns, and make informed decisions based on data-driven analyses.
4. **Model Evaluation:** Learners develop expertise in evaluating machine learning models using diverse metrics such as accuracy, precision, recall, specificity, and F1-score. They understand the importance of selecting appropriate evaluation metrics based on the project's objectives and domain-specific requirements.
5. **Interdisciplinary Knowledge:** The project encourages interdisciplinary learning by integrating domain knowledge of plant diseases, agricultural practices, and image analysis techniques with machine learning concepts. Participants gain a holistic understanding of how machine learning can be applied to real-world problems in agriculture and biology.
6. **Communication and Collaboration:** Through presenting findings, discussing results, and collaborating with peers or domain experts, participants enhance their communication skills. They learn to effectively communicate technical concepts, justify modeling decisions, and collaborate in multidisciplinary teams.
7. **Critical Thinking:** Engaging in result comparison, analysis, and interpretation fosters critical thinking skills. Participants learn to critically evaluate model performance, identify areas for improvement, and propose actionable recommendations based on data-driven insights.
8. **Ethical Considerations:** The project raises awareness of ethical considerations in machine learning, such as fairness, bias, and privacy. Participants learn to navigate ethical dilemmas

# CONCLUSION WITH CHALLENGES

Even though there are various methods for detecting and classifying plant diseases using automatic or computer vision, research into this field has been lacking. In addition, there are few commercial options, with the exception of those focusing on the identification of plant species via photographs. Over the last few years, there has been tremendous progress in the performance of convolutional neural networks. The new generation of convolutional neural networks (CNNs) has shown promising results in the field of image recognition. A novel approach to automatically classifying and detecting plant diseases from leaf images was examined through this project utilizing deep learning techniques. With an accuracy of 90%, the developed model could distinguish healthy leaves from eight diseases that could be observed visually. On the basis of this high level of performance, it becomes apparent that convolutional neural networks are highly suitable for automatic diagnosis and detection of plants

the plant disease detection project using machine learning has provided valuable learning outcomes and insights. Participants have developed technical proficiency in machine learning techniques, problem-solving skills, and a deeper understanding of data interpretation and model evaluation. The interdisciplinary nature of the project has fostered collaboration and communication skills

# FUTURE SCOPE

The main goal for the future project is to develop a complete system comprising a trained model on the server, as well as an application for mobile phones that display recognized diseases in fruits, vegetables, and other plants based on photographs taken from the phone camera. This application will aid farmers by facilitating the recognition and treatment of plant diseases in a timely manner and help them make informed decisions when utilizing chemical pesticides. Also, future work will involve spreading the use of the model across a wider land area by training it to detect plant diseases on aerial photos from orchards and vineyards captured with drones, in addition to convolution neural networks for object detection. Drones and other autonomous vehicles, such as smartphones, to be used for real-time monitoring and dynamic disease detection in large-scale open-field cultivations. A future possibility for agronomists working at remote locations could be the development of an automated pesticide prescription system that would require the approval of an automated disease diagnosis system to allow the farmers to purchase appropriate pesticides. Thus, the uncontrolled acquisition of pesticides could be severely restricted, resulting in their excessive use and misuse, with their potentially catastrophic effects environment.

# REFERENCES

[1] Chohan, Murk & Khan, Adil & Chohan, Rozina & Hassan, Muhammad. (2020). Plant Disease Detection using Deep Learning. International Journal of Recent

Technology and Engineering. 9. 909-914. 10.35940/ijrte.A2139.059120.

[2] Mohanty SP, Hughes DP. 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.

[3] Vijeta Shrivastava, Pushpanjali, Samreen Fatima, Indrajit Das, " Plant leaf diseases detection and classification using machine learning", International

Journal of Latest Trends in Engineering and Technology , Vol.10, Issue.2, April 2018.

[4] Ferentinos, Konstantinos. (2018). “Deep learning models for plant disease detection and diagnosis. Computers and Electronics in Agriculture.” 145. 311-318.

10.1016/j.compag.2018.01.009.

[5] Guo, Yan, et al. “Plant Disease Identification Based on Deep Learning Algorithm in Smart Farming.” Discrete Dynamics in Nature and Society, Hindawi, 18

Aug. 2020.

[6] Arsenovic Marko, Karanovic Mirjana, Sladojevic S. “Solving Current Limitations of Deep Learning Based Approaches for Plant Disease

Detection.” Symmetry. 2019; 11(7):939.

[7] B. Sandika, S. Avil, S. Sanat and P. Srinivasu. "Random forest based classification of diseases in grapes from images captured in uncontrolled

environments," 2016 IEEE 13th International Conference on Signal Processing (ICSP), 2016, pp. 1775-1780, doi: 10.1109/ICSP.2016.7878133.

[8] Alagumariappan P, Dewan NJ, Muthukrishnan GN, Raju BKB, Bilal RAA, Sankaran V. Intelligent Plant Disease Identification System Using Machine

Learning. Engineering Proceedings. 2020; 2(1):49

[9] Adrian Rosebrock. “Keras ImageDataGenerator and Data Augmentation PyImageSearch.” PyImageSearch, 8 July 2019.

[10] MK Gurucharan. “Basic CNN Architecture: Explaining 5 Layers of Convolutional Neural Network | UpGrad Blog.” UpGrad Blog, 7 Dec. 2020.

[11] Michael A. Nielsen. “Neural Networks and Deep Learning.” Neural Networks and Deep Learning.

[12] DeepAI. “Adam Definition | DeepAI.” DeepAI, DeepAI, 5AD.

[13] Christian Versloot. “Sparse Categorical Crossentropy Loss with TF 2 and Keras – Machine Curve.” Machine Curve, 6 Oct. 2019.

[14] Abhinavsagar. “GitHub – Abhinavsagar/Plant-Disease : Code for the Paper On Using Transfer Learning for Plant Disease Detection.” GitHub

[15] vipool. “Plant Diseases Classification Using AlexNet | Kaggle.” Kaggle: Your Machine Learning and Data Science Community, Kaggle, 29 Nov. 2018.