A Course Based Project Report on

CROP DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

Submitted to the

Department of Information Technology

in partial fulfillment of the requirements for the completion of course

DEEP LEARNING APPLICATIONS (22PC2DS402)

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

Submitted by

ATTELLI SANJAY KUMAR	22071A12D3
KAMINENI SAI SRI RAM	22071A12F0
N AADITYA SANTOSH	22071A12G7
A AKHANDA JABILI	22071A12D2

Under the guidance of

Mrs. J. SRAVANTHI (Course Instructor)

Department of IT, VNRVJIET

TAMASOMA JYOTIRGAMAYA

Estd.1995

DEPARTMENT OF INFORMATION TECHNOLOGY

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

NOV 2025

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as
“College with Potential for Excellence” by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY

CERTIFICATE

This is to certify that the project report entitled “**CROP DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)**” is a bonafide work done under our supervision and is being submitted by **A. SANJAY KUMAR (22071A12D3), K. SAI SRI RAM (22071A12F0), N. AADITYA SANTOSH (22071A12G7), A. AKHANDA JABILI (22071A12D2)** in partial fulfillment for the award of the degree of Bachelor of Technology in Information Technology, of the VNRVJIET, Hyderabad during the academic year 2024-2025.

Mrs J.SRAVANTHI	**Dr. N. MANGATHAYARU**
Assistant Professor, IT	Professor - HOD, IT

NOV 2025

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade,

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We declare that the course based project work entitled “**CROP DISEASE DETECTION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)**” submitted in the Department of Information Technology, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Information Technology** is a bonafide record of our own work carried out under the supervision of **MRS. J. SRAVANTHI (ASSISTANT PROFESSOR) , DEPARTMENT OF IT, VNRVJIET.** Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

SANJAY KUMAR 22071A12D3	SAI SRI RAM 22071A12F0	N. AADITYA SANTOSH 22071A12G7
A. AKHANDA JABILI 22071A12D2		

NOV 2025

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, Sri. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr. C.D Naidu, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor Dr. N. MANGATHAYARU Professor and Head, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, Mrs. J. Sravanthi , Assistant Professor in Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

A. SANJAY KUMAR	22071A12D3
SAI SRI RAM	22071A12F0
N . AADITYA SANTOSH	22071A12G7
A. AKHANDA JABILI	22071A12D2

NOV 2025

ABSTRACT

Crop Disease Prediction Using Convolutional Neural Networks (CNN) addresses the critical need for efficient and accurate crop disease detection in agriculture. The system leverages the power of deep learning, specifically CNNs, to automate the identification of crop diseases from images of plant leaves. The objective is to create a deep learning model capable of classifying leaf images into various categories, including healthy and diseased states (e.g., early blight, late blight, bacterial spot), providing valuable assistance to farmers and agronomists for early disease detection. The methodology involves utilizing the publicly available PlantVillage dataset, a rich resource containing thousands of labeled images of healthy and diseased plant leaves across multiple species. Preprocessing steps include image resizing to 256x256 pixels and normalization of pixel values. A CNN architecture, implemented using TensorFlow/Keras, is employed. This architecture consists of convolutional layers for feature extraction, maxpooling layers for dimensionality reduction, dropout layers to mitigate overfitting, and fully connected dense layers for classification. The model is trained using a split dataset, with performance evaluated. The resulting system offers real-time disease prediction from leaf images, achieving high accuracy through deep learning. Potential applications extend to assisting farmers in early disease diagnosis, integrating into agricultural monitoring systems, and enhancing smart farming and precision agriculture tools.

CONTENTS

1. INTRODUCTION

Crop diseases significantly impact global food security by reducing yields and causing economic losses for farmers. Early and accurate detection is essential for timely intervention. This project proposes a deep learning-based solution using Convolutional Neural Networks (CNNs) to automatically identify plant diseases from leaf images. Leveraging the PlantVillage dataset, which contains thousands of labeled images of healthy and diseased leaves, the system classifies conditions such as blight and bacterial spots. Preprocessing techniques like image resizing, normalization, and data augmentation are applied to improve model accuracy and generalization.

A custom CNN model is developed using TensorFlow/Keras, incorporating convolutional and pooling layers for feature extraction, along with dropout layers to minimize overfitting. The model is evaluated using key metrics including accuracy, precision, recall, and confusion matrix. The final system delivers fast, reliable predictions, offering practical support for farmers in disease management. Its scalability makes it adaptable to various crops and diseases, contributing to precision agriculture and promoting sustainable farming practices worldwide.

OBJECTIVES:

Dataset Handling

- Load the PlantVillage dataset.
- Extract images from a zip archive.
- Convert images to NumPy arrays.
- Resize images to a consistent size (256x256).
- Normalize pixel values for improved model training.

Data Preprocessing

- Create a numerical representation of labels.
- One-hot encode labels for categorical classification.
- Split data into training, validation, and testing sets.

Model Building and Training

- Construct a sequential CNN model using Keras.
- Define convolutional layers for feature extraction.
- Include max-pooling layers to reduce dimensionality.
- Add dropout layers to prevent overfitting.
- Use a dense layer for classification.
- Compile the model with categorical cross-entropy loss and Adam optimizer.
- Train the model using the training and validation datasets.
- Save the trained model for future use.

Model Evaluation

- Evaluate model performance on the test dataset.
- Report accuracy.
- Plot model accuracy and validation accuracy over epochs.
- Deployment using Streamlit.

Title	Authors	Year	Technology Used	Summary	Link
Rice Diseases Classification Using Feature Selection and Rule Generation Techniques [3]	Phadikar, S., Sil, J.	2013	Support Vector Machines (SVM), Feature Selection, Rule Generation	Focuses on detecting and classifying rice diseases using SVM along with rule-based classification and selected features for improved precision.	ScienceDirect
Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification[4]	Sladojevic, S., Arsenovic, M., et al.	2016	DeepNeuralNetworks (DNN),Convolutional Neural Networks (CNN)	Proposes a deep learning model for recognizing plant diseases from leaf images, achieving high accuracy in classifying multiple disease types using DNN.	ResearchGate
Plant Disease Detection Using Convolutional Neural Network [5]	Srivastava, P., Mishra, K., Awasthi, V., Sahu, V. K., Pal, P. K.	2021	Image Processing	Utilizes CNNs for automatic detection of plant diseases from leaf images, demonstrating high accuracy and efficiency in classification tasks.	International Journal of Advanced Research
Plant Disease Detection Using Hybrid Model Based on Convolutional Autoencoder and Convolutional Neural Network [1]	Punam Bedi	2021	Convolutional Autoencoder (CAE), Convolutional Neural Network (CNN)	Proposes a hybrid model combining CAE and CNN for automatic plant disease detection, achieving high accuracy with reduced training parameters.	ScienceDirect
Multi-Class PlantLeaf Disease Detection: A CNN-based Approach with Mobile Applicatio[2]	MdAziz Hosen Foysal1 , Foyez Ahmed2 , Md Zahurul Haque3	2023	Convolutional Neural Networks (CNN), Mobile Application	Investigates advanced techniques in plant disease detection, includes methods, and mobile technologies. Highresolution images of plant leaves were analyzed using CNNs to detect symptoms of various diseases, such as blight, mildew, and rust.	arXiv

3. SYSTEM REQUIREMENTS

3.1 FUNCTIONAL REQUIREMENTS

1. Allow users to upload crop leaf images for disease detection.
2. Preprocess images and analyze them using a trained CNN model to identify crop diseases.
3. Display the predicted disease (or healthy status).

3.2 NON-FUNCTIONAL REQUIREMENTS

The system should be designed to ensure high performance, scalability, and robustness, capable of processing large datasets in real-time with minimal latency. It must be reliable, maintaining accuracy and consistency across varying environments and file types.

Non-Functional Requirements:

1. Accuracy: The system should deliver reliable disease predictions with a high accuracy rate.

2. Performance: The system should generate results quickly, ideally within a few seconds of image upload.

3. Usability: The interface should be user-friendly and accessible, even for users with limited technical knowledge.

4. Scalability: The system should handle an increasing number of users and image uploads without performance issues.

5. Security: User data and uploaded images must be securely handled and protected from unauthorized access.

3.3 SOFTWARE REQUIREMENTS:

1. Programming Language: Python

2. Libraries and Frameworks: TensorFlow (for building and training the CNN model), OpenCV (for image processing), NumPy and Pandas (for data handling), Pillow (for image loading and manipulation).

3. Algorithms: Convolutional Neural Networks (CNN) for image classification and disease detection

3.4 HARDWARE REQUIREMENTS:

1. **Processor**: Intel i5 or higher (for handling image processing and CNN-based inference tasks)
2. **RAM**: Minimum 8 GB (16 GB or higher recommended for smooth performance during model training and large image dataset handling)
3. **Storage**: 256 GB SSD or higher (for storing crop images, model files, and result reports with fast read/write access)
4. **Graphics Processing Unit (GPU)**: NVIDIA GTX 1060 or higher (optional, but recommended for accelerating CNN model training and improving prediction speed)
5. **Network**: Stable internet connection (required for accessing online datasets, cloud-based deployment, and remote updates)

4. DATASET

4.1 SOURCE

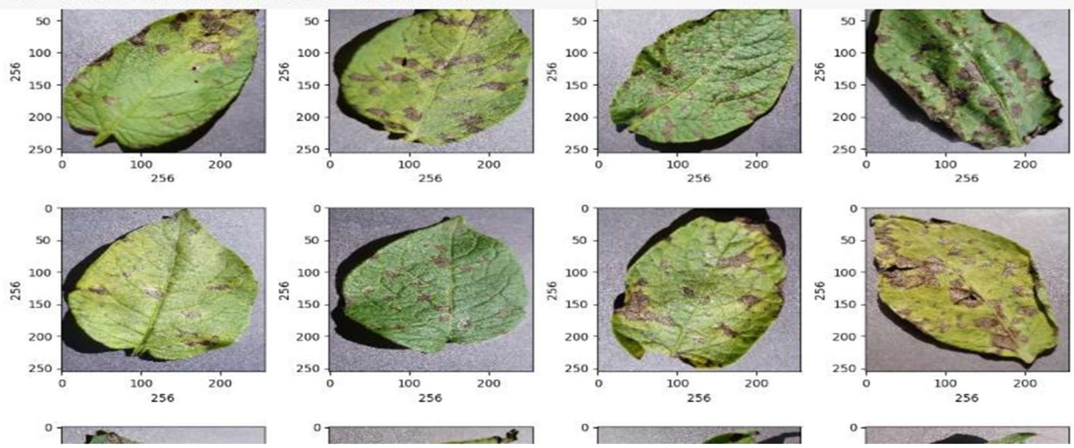
Dataset source: **https://www.kaggle.com/datasets/emmarex/plantdisease/data**

4.2 DESCRIPTION

The dataset used here is a subset of PlantVillage dataset which is a comprehensive collection of images depicting healthy and diseased plant leaves, accompanied by corresponding labels. This dataset is widely utilized in the development and evaluation models for plant disease detection and classification.

```
plt.figure(figsize = (12, 12))
dataset_path = "/content/plant_data/PlantVillage/Potato___Early_blight"

for i in range(1, 17):
    plt.subplot(4, 4, i)
    plt.tight_layout()
    rand_img = imread(dataset_path +'/'+ random.choice(sorted(os.listdir(dataset_path))))
    plt.imshow(rand_img)
    plt.xlabel(rand_img.shape[1], fontsize = 10) # width of image
    plt.ylabel(rand_img.shape[0], fontsize = 10) # height of image
```



There are a total of 15 classes in the dataset but for computational constraints we have used only 8 classes.

```
all_labels = ['Potato___healthy', 'Tomato_Early_blight', 'Tomato_Bacterial_spot', 'Tomato_healthy',
              'Pepper__bell___healthy', 'Potato___Late_blight',
              'Pepper__bell___Bacterial_spot','Potato___Early_blight']
```

4.3 METHODOLOGY

The methodology for developing the crop disease detection system involves several key steps: first, collecting a comprehensive dataset of crop images, including both healthy and diseased leaves across various crop types. Next, the data is preprocessed by resizing images, normalizing pixel values, and augmenting the dataset to improve model generalization. Feature extraction and classification are performed using a Convolutional Neural Network (CNN) architecture. The model is trained and finetuned on the preprocessed image dataset. Various evaluation metrics, including accuracy, precision, recall, and F1-score, are used to assess the model's performance and identify the best-performing configuration for reliable disease classification.

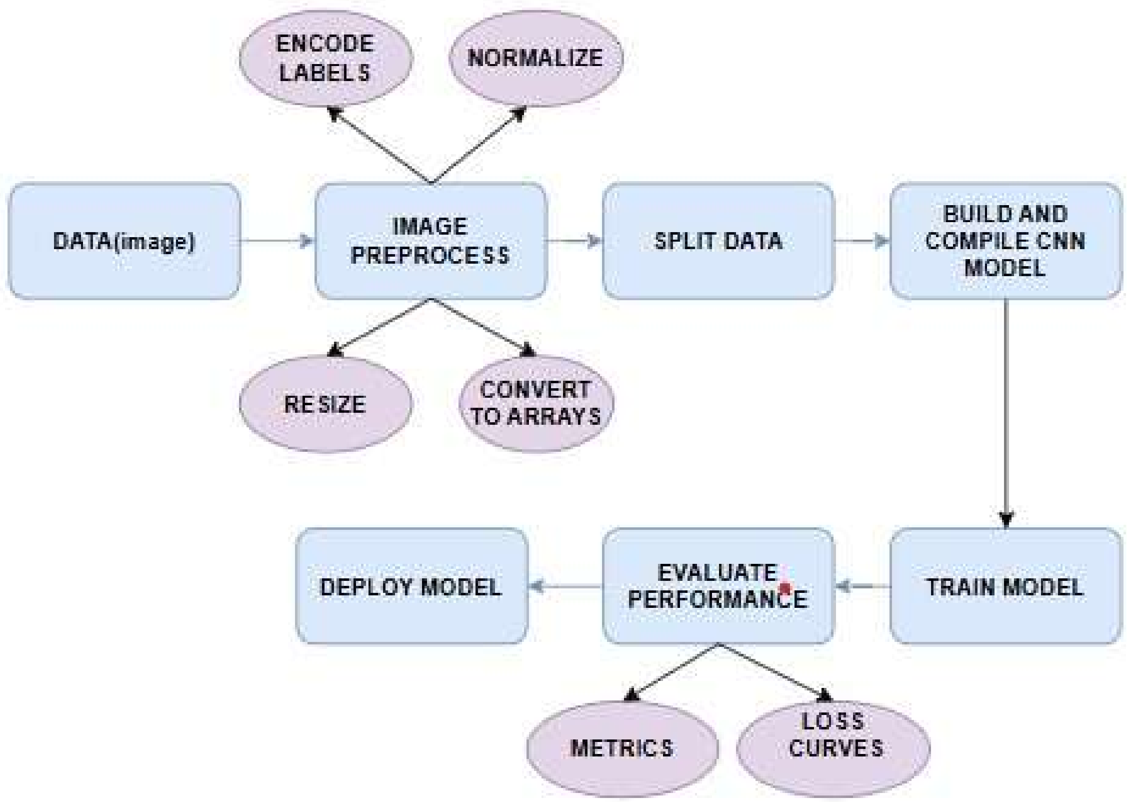


Fig: Work flowchart of the system where image is provided as input

5. IMPLEMENTATION

There are 3 steps:

5.1 Image Preprocessing

5.2 Building CNN model

5.3 Training and Evaluation

5.1. IMAGE PREPROCESSING:

This part of the code is preparing the images so that they are all in the same format and size before being used for training a model. It goes through each image, makes sure it's in color (with three channels), resizes it to 256 by 256 pixels, and changes its format so the model can understand it. After that, the pixel values are adjusted to be between 0 and 1, which helps the model learn better. After resizing the shapes if x_train and x_test are: **x_train shape: (6541, 256, 256, 3) x_test shape: (2804, 256, 256, 3)**

```
from PIL import Image
import numpy as np

def resize_images(image_list, target_size=(256, 256)):
    resized = []
    for img in image_list:
        # Convert float32 arrays to uint8 (if needed)
        if isinstance(img, np.ndarray) and img.dtype != np.uint8:
            img = (img * 255).astype(np.uint8)

        if isinstance(img, np.ndarray):
            img = Image.fromarray(img)

        img = img.convert("RGB")  # Ensure 3 channels
        img = img.resize(target_size)
        resized.append(np.array(img))
    return np.array(resized, dtype=np.float32)

# Resize and convert
x_train = resize_images(x_train)
x_test = resize_images(x_test)

# Normalize
x_train /= 255.0
x_test /= 255.0
```

Fig: Image resizing and normalizing the pixels

The label in y_train and y_test are converted into numpy arrays and then one hot encoded to get the proper dimensions for the model for predictions. Here 8 represents number of classes or the number of nodes in the output layer.

```python
from tensorflow.keras.utils import to_categorical

# One-hot encode labels
y_train = to_categorical(y_train)
y_test= to_categorical(y_test)

print("y_train_cat shape:", y_train.shape)
print("y_test_cat shape:", y_test.shape)
```

```
y_train_cat shape: (6541, 8)
y_test_cat shape: (2804, 8)
```

Fig: One-hot encoding the labels

5.2 BUILDING CNN MODEL

To detect crop diseases, a Convolutional Neural Network (CNN) was developed using a sequential architecture. The model consists of three convolutional blocks, each containing a convolutional layer with increasing filter sizes (32, 64, and 128) followed by a max-pooling layer to reduce spatial dimensions while preserving important features. All convolution layers use the ReLU activation function and padding to maintain input dimensions. After feature extraction, the output is flattened and passed through a fully connected dense layer with 256 neurons and ReLU activation. To reduce overfitting, a dropout layer with a rate of 0.5 is added. Finally, the output layer uses a softmax activation function with 8 units, corresponding to the number of crop disease classes. The model is compiled using the Adam optimizer with a learning rate of 0.0001 and trained using categorical crossentropy loss, with accuracy as the evaluation metric

```python
model = Sequential()

# 1st Conv Block
model.add(Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=(256, 256, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))

# 2nd Conv Block
model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# 3rd Conv Block
model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))

# Fully Connected Layers
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(8, activation='softmax'))
```

Fig: CNN model with alternate Convolutional and pooling layers with dense layers at the end for classification

5.3 TRAINING AND EVALUATION

There are a total of 9345 images present out of which 6541 images are used for training and remaining 2804 are used for testing the model. The training set was again split into training and validation set and the model was trained for 12 epochs due to computational constraints and then evaluated on the validation set. The following image shows the training and validation accuracy curves for the 12 epochs.

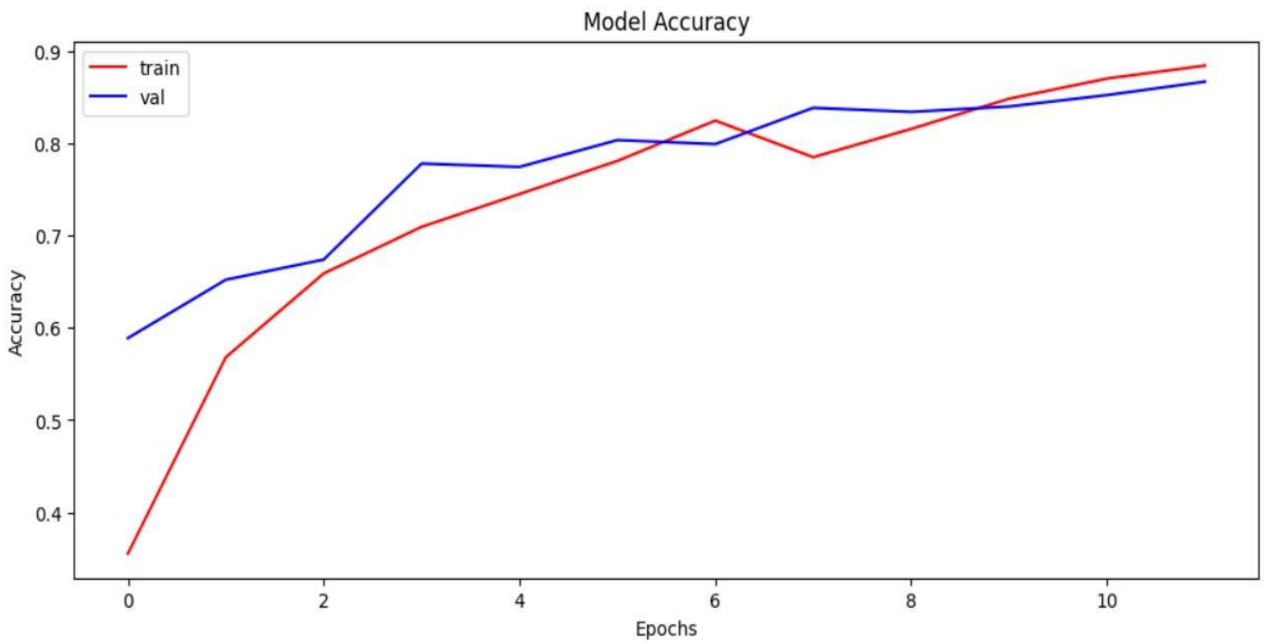


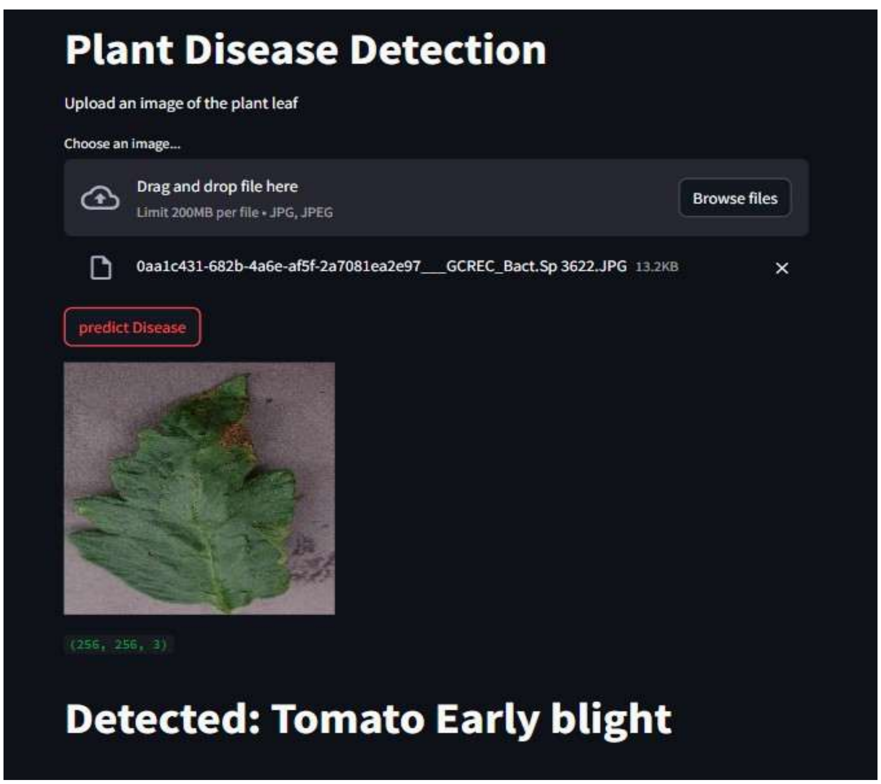
Fig: The plot of training accuracy vs epochs and validation accuracy vs epochs

6. RESULTS

```
[ ]   1 print("Calculating model accuracy")
      2
      3 scores = model.evaluate(x_test, y_test)
      4 print(f"Test Accuracy: {scores[1] * 100}")

Calculating model accuracy
88/88 ━━━━━━━━━━━━━━━━━━━━ 79s 876ms/step - accuracy: 0.8500 - loss: 0.4039
Test Accuracy: 86.09129786491394
```

The model trained for crop disease detection using a Convolutional Neural Network (CNN) achieved an **accuracy 86.09% on the test data**. Accuracy, reflects the proportion of correctly classified images out of the total number of predictions made by the model. Additionally, the relatively low loss value of **0.4039** suggests that the model has learned relevant patterns without significant overfitting, making it a reliable tool for practical deployment in agricultural settings to support early disease detection and informed decision-making.



This is the GUI that enables the user to upload the image of the plant leaf and detect whether it has been infected by any disease or not. It is built using **streamlit,** a python GUI framework. The GUI displays the label of the detected disease.

7. CONCLUSION AND FUTURE SCOPE

This project successfully developed a Convolutional Neural Network (CNN) model for accurate crop disease prediction using the PlantVillage dataset. The model effectively classified leaf images into various disease categories and healthy states, achieving strong performance across accuracy, precision, and recall metrics. These results highlight the potential of CNNs in automating plant disease diagnosis, offering a faster and more reliable alternative to traditional manual methods.

The system's early detection capabilities can help reduce crop losses, boost yields, and support farmers' livelihoods, especially in disease-prone regions. Its adaptability to different crops and diseases enables broader application in diverse agricultural settings. While the model shows promising results, future work could focus on enhancing robustness with more diverse data and exploring advanced techniques like transfer learning. Integrating the solution into user-friendly platforms, such as mobile apps, could further increase its real-world impact.

8. BIBLIOGRAPHY

1. Punam Bedi and Pushkar Gole. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. Artificial Intelligence in Agriculture, 5:90–101, 2021.

2. Md Aziz Hosen Foysal, Foyez Ahmed, and Md Zahurul Haque. Multi-class plant leaf disease detection: A cnn-based approach with mobile app integration. arXiv preprint arXiv:2408.15289, 2024.

3. Santanu Phadikar, Jaya Sil, and Asit Kumar Das. Rice diseases classification using feature selection and rule generation techniques. Computers and electronics in agriculture, 90:76–85, 2013.

4. Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016(1):3289801, 2016.

5. Prakanshu Srivastava, Kritika Mishra, Vibhav Awasthi, Vivek Kumar Sahu, and Pawan Kumar Pal. Plant disease detection using convolutional neural network. International Journal of Advanced Research, 9(01):691–698, 2021.

6. https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
7. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html
8. https://viso.ai/computer-vision/image-classification/