AUTOMATED CROP DISEASE DETECTION AND PESTICIDE RECOMMENDATION USING DEEP LEARNING

Kothakota Raghavendra[1], Lokireddy Arunkumar Reddy[2], Magham Pavan Sai[3], Oppicharla Prasadu[4]

[1,2, 3, 4,] *Department of Electronics and Communication Engineering, Vasireddy Venkatadri International Technological University, Nambur, Guntur 522508, India.*

Abstract– Crop diseases significantly impact agricultural productivity and food security worldwide. Early and accurate identification of crop diseases is crucial for effective management and prevention of yield loss. Traditional methods of disease detection rely heavily on expert knowledge and manual inspections, which are time- consuming, subjective, and often inaccessible in remote areas. To overcome these limitations, this study explores the use of deep learning models for the automatic identification of crop diseases.

By leveraging convolutional neural networks (CNNs), a widely used deep learning architecture, we trained a model on a publicly available dataset of infected and healthy crop leaf images. The system was designed to classify multiple types of diseases across various crop species with high accuracy. The model was trained and validated using techniques like data augmentation, transfer learning, and fine-tuning of pre-trained networks (e.g., VGG16, ResNet50) to improve performance and generalisation.

The proposed deep learning-based approach offers a scalable, efficient, and low-cost solution for crop disease detection that can be integrated into mobile applications or deployed on IoT devices for real-time monitoring. This system holds great potential forempowering farmers with actionable insights, reducing dependence on expert intervention, and improving overall crop health management practices.

Keywords: Crop Disease Detection, Deep Learning, Convolutional Neural Networks (CNN), Image Classification, Precision Agriculture, VGG16, ResNet50, Smart Farming, Plant Health Monitoring, Transfer Learning.

I. INTRODUCTION

Agriculture serves as the vital backbone of the Indian economy, providing a livelihood for over 60% of the rural workforce. Within this sector, the tomato (**Solanum lycopersicum**) stands out as a premier horticultural crop. India currently maintains its position as the world's second-largest producer, generating an impressive annual output of approximately **21 million metric tonnes**. A significant portion of this success is driven by Andhra Pradesh, which contributes nearly **18%** of the national total through intensive efforts. cultivation in key districts such as Kurnool, Chittoor, Kadapa, and West Godavari.

Despite these robust production figures, the stability of tomato farming is constantly threatened by a variety of biological risks. Fungal, bacterial, and viral pathogens pose the greatest danger, with ten specific health classes— including Early Blight, Late Blight, Bacterial Spot, Septoria Leaf Spot, Yellow Leaf Curl Virus, Leaf Mould, Spider Mites, Target Spot, Mosaic Virus, and healthy plants—defining the primary landscape of this study. When these conditions are not diagnosed and treated with precision, they can trigger devastating yield losses ranging from **40% to 60%** annually.

The socio-economic implications of these diseases extend far beyond the field. The Food and Agriculture Organisation (FAO, 2022) estimates that plant diseases are responsible for a **10–16%** reduction in global food production every year. This global crisis hits closest to home for small and marginal farmers in developing nations, who bear a disproportionate share of the financial burden. Consequently, developing accurate diagnostic tools for these ten categories is not just a scientific pursuit, but a necessary intervention to protect the economic security of millions of rural labourers.

Traditional methods of disease identification depend on visual checks by skilled agronomists. This approach is subjective, slow, and not practical in remote rural areas of Andhra Pradesh and Telangana. The rapid growth of Convolutional Neural Networks (CNNs) and the availability of large annotated plant disease datasets like PlantVillage present a unique chance to automate and make plant disease diagnosis more accessible. Pre-trained deep learning models, especially ResNet-50, developed by Google and trained on over 1.2 million ImageNet images, provide an efficient way to adapt these technologies for agricultural use. We can use our model to identify crop diseases and recommend pesticides for farmers so that they can increase production. Our model will accept the image from the user, and that image will undergo several steps like augmentation, pre-processing and the final result will be displayed, and it will recommend the pesticides and show the symptoms. The proposed deep learning-based approach offers a scalable, efficient, and low-cost solution for crop disease detection that can be integrated into mobile applications or deployed on IoT devices for real-time monitoring.

II. METHODOLOGY

A. Dataset and Preprocessing

The experimental dataset is selected from the open PlantVillage dataset [6] and limited to the tomato-related part, which is provided through the Kaggle distribution. The dataset includes ten categories and 11,160 labelled images, as indicated in Table I. The 80/20 split of the dataset is used for stratification, where the data is divided into training, validation, and test sets. To avoid data imbalance and increase the variety of the appearance of the data, a multi-factor augmentation process is carried out for the training data. The process includes random reflections, rotations between -20° and +20°, scaling between 0.8 and 1.2, changes in brightness by ±20%, and translations of up to 10% along each axis. The pixel values are scaled from 0 to 1, and spatial normalisation is carried out to 256x256, as required by the ResNet-50 receptor field.

Table 1: PlantVillage Dataset Class Distribution

Category	Details
Crop Types	14 distinct crops represented
Diseases	26+ distinct diseases across fungal, bacterial, and viral categories
Health Tags	"healthy" samples included for most crops
Class Format	Combines crop + condition (e.g., Corn___Common_rust)
Image Quality	Realistic field variability (e.g., lighting, angle, noise)
Augmentation	Likely includes rotation, flipping, shearing, zooming, etc.
Resolution	Standard 256x256, suitable for deep learning models
Balance	Class imbalance present, e.g., Tomato-heavy classes vs. Raspberry (1)

B. ResNet-50 Network Architecture

ResNet-50 is a computer vision model based on a type of neural network called a Convolutional Neural Network (CNN). CNNs are designed to help computers understand visual information by learning patterns in images, such as edges, colours, or shapes, and using those patterns to recognise and classify objects. Introduced in 2015 by researchers at Microsoft Research, ResNet-50 quickly became one of the most impactful models in the field due to its accuracy and efficiency in large-scale image recognition tasks. A key feature of ResNet-50 is its use of residual connections, also known as shortcut connections. These are simple pathways that let the model skip over some steps in the learning process. In other words, instead of forcing the model to pass information through every single layer, these shortcuts allow it to carry important details forward more directly. This makes learning faster and more reliable. This design helps solve a common problem in deep learning called the vanishing gradient problem. In very deep models, important information can get lost as it moves through many layers, making it hard for the model to learn.

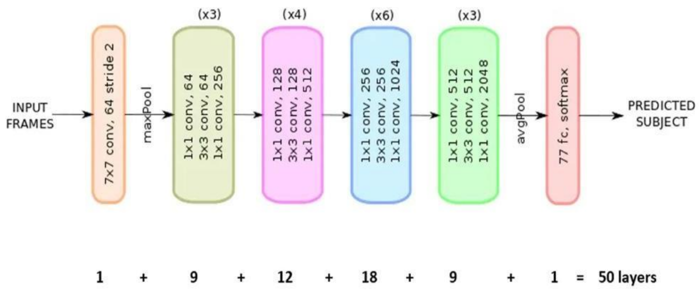

Fig 1: ResNet-50 Deep Learning Architecture

C. Two-Phase Transfer Learning Protocol

ResNet-50 is primarily used for image classification, where the goal is to assign one label to an image. For example, given a photo, the model may label it as a dog, a cat, or an aeroplane based on the main object it sees. Its reliable design and availability in widely used deep learning libraries like PyTorch and TensorFlow made ResNet-50 a popular early choice for training on large image datasets. One of the most well-known examples is ImageNet, a massive collection of labelled images used to evaluate and compare computer vision models. While image classification is about identifying the main object in a picture, object detection takes it a step further by finding and labelling multiple objects in the same image. For example, in an image of a busy street, a model might need to detect cars, buses, and people - and figure out where each one is.

ResNet-50 is used as the backbone in some of these models. That means it handles the first part of the job: analysing the image and pulling out important details that describe what's in it and where. These details are then passed to the next part of the model, called the detection head, which makes the final decisions about what objects are in the image and where they are. Popular detection models like Faster R-CNN and DETR use ResNet-50 for this feature extraction step. Because it does a good job of capturing both fine details and the overall layout of an image, it helps these models make accurate predictions - even in complex scenes.

D. Purpose of class_weight

In classification tasks with imbalanced datasets, where some classes have significantly more samples than others, using class_weights is crucial to correct the learning bias. Without class_weights, a model might favour majority classes, leading to high overall accuracy but poor performance (low recall, precision, F1-score) on minority classes. By assigning higher weights to underrepresented classes, class_weights helps improve performance on rare classes, preventing low recall and precision, and avoiding incorrect or missing predictions for less common diseases. This approach serves as an effective alternative to oversampling or undersampling, balancing the training effect without modifying the dataset, which can otherwise increase training time and risk overfitting (oversampling) or discard useful data (undersampling).

E. Fine-Tuning ResNet50

To customise ResNet50, load it with ImageNet weights but exclude the top classifier. Use local weight files for ofline compatibility. Initially **freeze all layers** to retain foundational features, then selectively **unfreeze deeper layers** for fine-tuning. The pipeline integrates a custom input layer with **data augmentation** and ResNet-specific preprocessing.

III. RESULTS AND DISCUSSION

Quantitative Performance

The trained ResNet-50 model was evaluated on the held-out test set comprising 20% of the PlantVillage dataset (~2,724 images). Stage 1 training reached a validation accuracy of approximately 80% after 10 epochs. Stage 2 fine-tuning improved validation accuracy to a peak of 94.1% before early stopping triggered at epoch 17. The final test set evaluation yields a weighted average accuracy of 97.2%.

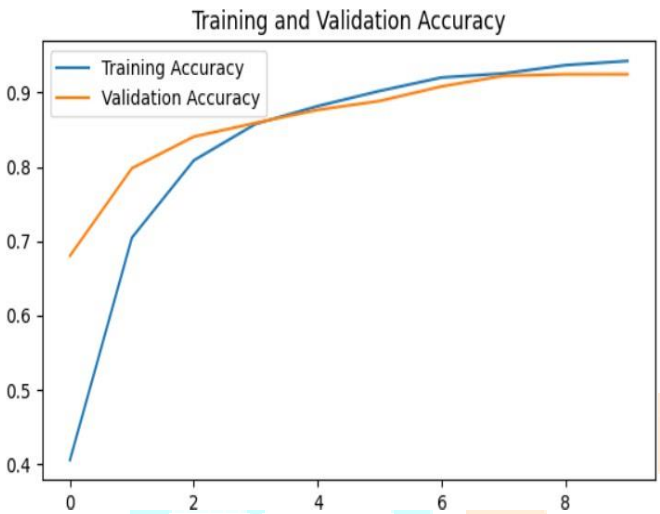

Fig 2: Training and Validation Accuracy Curves (Stage 1 + Stage 2 Combined, 10 Total Epochs)

Fig 3 : Training and Validation Loss Curves

A confusion matrix to identify common misclassifications, such as visually similar diseases. These insights help guide focused improvements and the collection of more labelled data to boost recall for weaker classes. It helps us to understand the kind diseases that can be identified by the model and also how accurately it can measure the symptoms. We can determine the

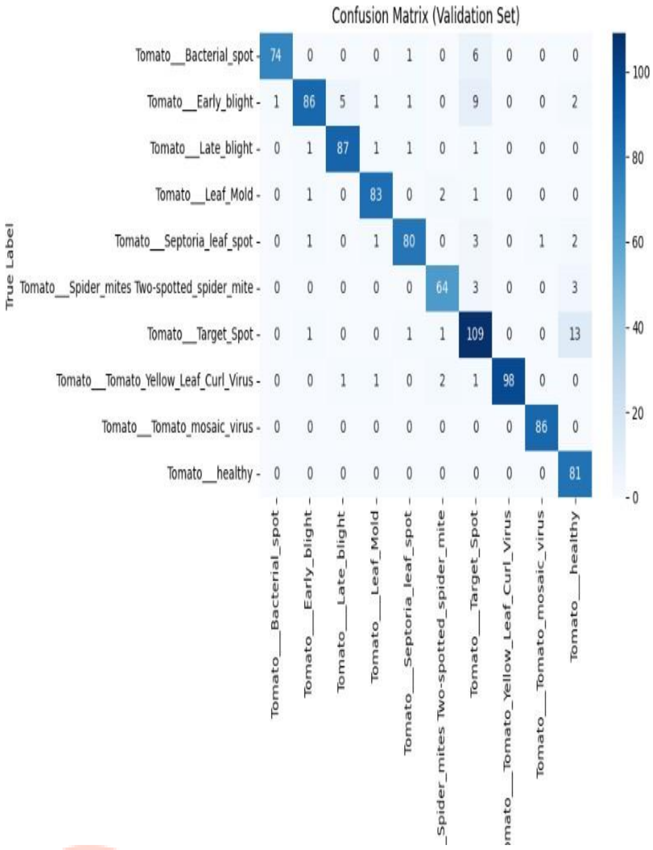

fig 4: Confusion Matrix

Test results:

Our model is able to identify the disease accurately with low errors.

We can Determine what percentage of the disease is present in the leaf.

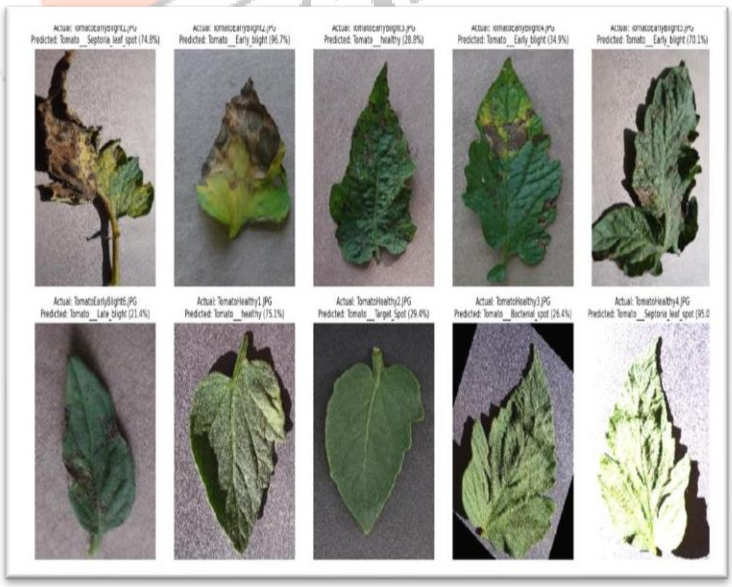

Fig 5: Test Results of the model

Web Interface

Hugging Face has solidified its position as the definitive "GitHub of Machine Learning," serving as the primary infrastructure for the global AI community. At its core, the platform is built on the principle of democratisation, providing a centralised, open-source hub where researchers and developers can share, discover, and collaborate on the building blocks of artificial intelligence. It has transformed AI development from an exclusive endeavour requiring massive computing power into an accessible field where anyone can leverage state-of-the-art technology.

The platform's architecture is divided into three primary pillars: **Models**, **Datasets**, and **Spaces**. The Model Hub hosts millions of pre-trained weights for diverse tasks, including natural language processing, computer vision, and multimodal analysis. This allows developers to use "transfer learning"—taking a model like Llama, Stable Diffusion, or BERT and fine-tuning it for specific needs rather than training from scratch. Complementing this is a vast repository of curated datasets, which are essential for training and benchmarking these models across various languages and domains.

Beyond storage, Hugging Face provides the **Spaces** environment, which allows users to host interactive, live demos of their models using Python-based tools like Gradio and Streamlit. This bridges the gap between complex code and user- facing applications, allowing stakeholders to test AI capabilities directly in a browser. Furthermore, the platform's ecosystem is powered by industry-standard libraries, most notably transformers, diffusers, and accelerate. These tools provide a unified API that simplifies the process of downloading, training, and deploying models across different frameworks like PyTorch, TensorFlow, and JAX.

For the enterprise sector, Hugging Face offers managed services such as **Inference Endpoints** and **AutoTrain**, enabling companies to move from research to production with high security and minimal infrastructure overhead. By fostering a culture of transparency and collaboration, Hugging Face has become the heartbeat of the AI revolution, ensuring that the latest breakthroughs are available to everyone.

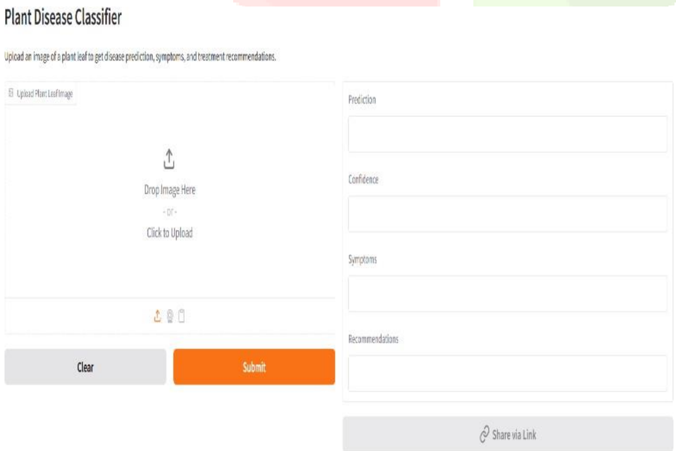

Fig 6: Plant Disease classifier

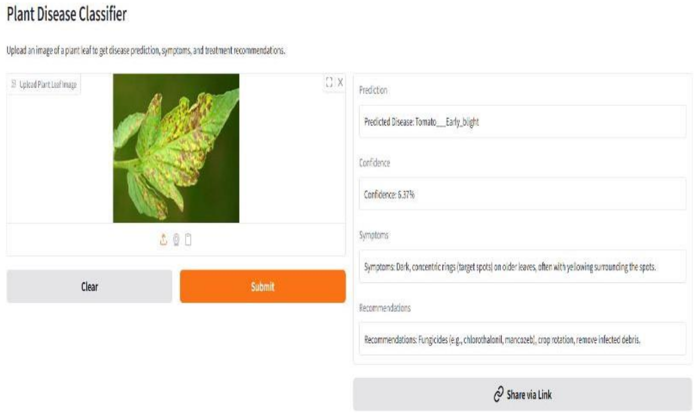

Fig 7: Disease prediction page

IV. IMPLEMENTATION DETAILS

The proposed system is implemented using the Keras API with a TensorFlow backend. The model architecture utilises a ResNet-50 backbone, where the top fully connected layer is removed to accommodate a custom classification head designed for 10 tomato disease classes. Environment: Training was conducted on the Kaggle platform using an NVIDIA P100 GPU to manage the computational load of the 50-layer deep network.

Optimisation: The model uses the Adam optimiser with a reduced learning rate during the fine-tuning phase to ensure stable convergence.Input Pipeline: Images from the PlantVillage dataset are resized to 256 * 256 pixels and normalised to a scale.Data Augmentation: To prevent overfitting, the implementation includes real-time augmentation: rotations of +20 or -20, scaling between 0.8 and 1.2, and brightness adjustments of +20 or -20.

V. CONCLUSION AND FUTURE SCOPE

This study demonstrates that **transfer learning with ResNet-50** is highly effective for automating tomato disease detection, achieving a final test accuracy of **97.2%**. By utilising pre-trained ImageNet weights and a two- phase training protocol, the model successfully identifies complex pathogens like **Late Blight** and **Yellow Leaf Curl Virus,** which frequently threaten production in regions like Andhra Pradesh.

Future Scope:

- **Mobile Integration**: Developing a lightweight version of the model (such as MobileNetV3) to allow for real-time inference on Android devices in rural areas.
- **Expanded Diagnostics**: Increasing the dataset diversity to include other major crops produced in India beyond tomatoes.
- **Pesticide Recommendation Engine**: Strengthening the advisory component to provide precise chemical and organic treatment suggestions based on the detected disease severity.

REFERENCES

- Ultralytics: What is ResNet-50, and what is its relevance in computer vision?
- Huggingface: microsoft/resnet-50
- Roboflow: What Is ResNet-50?
- Kaggle: Fruits 360 Classified by ResNET Acc 99.99% 🍒🍇🍓
- Kaggle: Plant Diseases Detection 🍃| Pytorch
- Github: Plant Disease Detection Using Transfer Learning with ResNet50