Plant Friend: Final Report

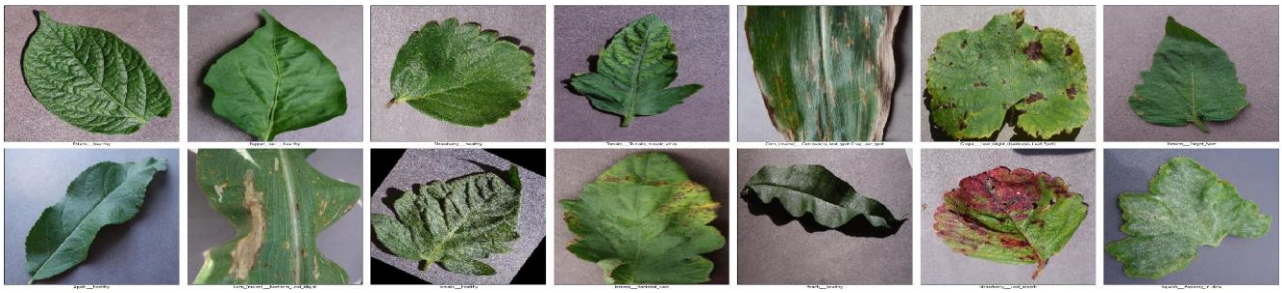
Section B | Team 6 | Mahin Rahman, Denisse Castillo, Siying Ren, Qingci Jiang, Yao Mu

Introduction and Motivation

Agriculture faces persistent challenges due to plant diseases, which can significantly hinder crop yield, reduce farmers' profitability, and impact global food security. Plant diseases often remain undetected until it is too late, causing economic losses and contributing to food scarcity. In fact, **plant diseases cause approximately $220 billion in annual losses globally as per the US Department of Agriculture**. Leveraging advances in computer vision and deep learning, our project, Plant Friend, aims to build a robust solution for plant disease detection. Using image classification techniques, the platform can identify diseases from leaf images and suggest appropriate remedies. This can help us establish a subscription platform from where we can earn revenue by helping farmers to take timely corrective actions against diseases.

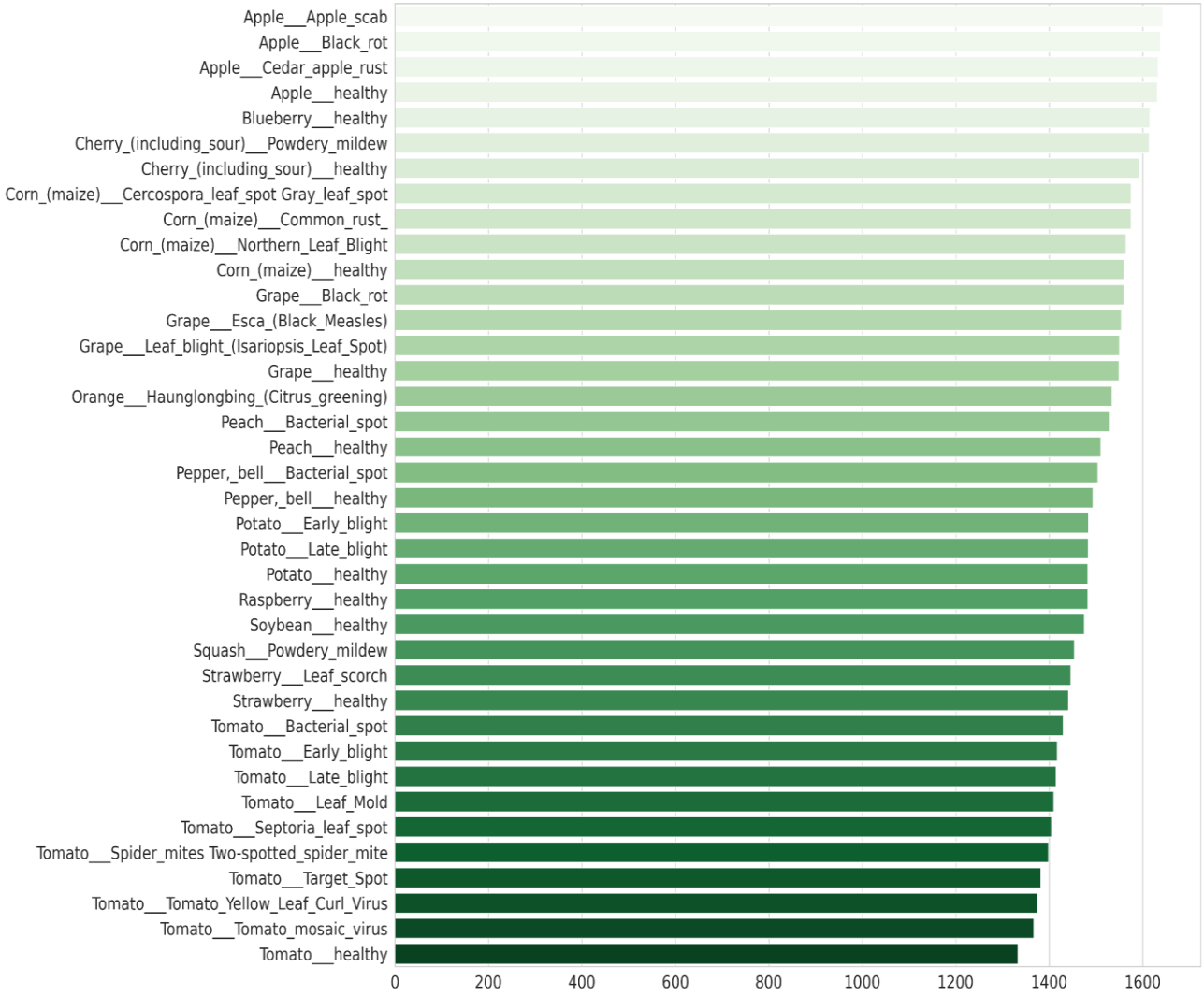
Data Understanding

The dataset used for this project is sourced from Kaggle's 'New Plant Diseases Dataset' and contains a total of 105,076 labeled images, representing 38 distinct classes. Each image corresponds to a diseased or a healthy plant state and data augmentation techniques were already applied on them. However, the imbalance in dataset sizes (70,295 training samples, 17,572 validation samples, and only 33 test samples) was concerning because a tiny test set cannot provide a reliable evaluation of the model's performance or generalization ability.



Data Preparation

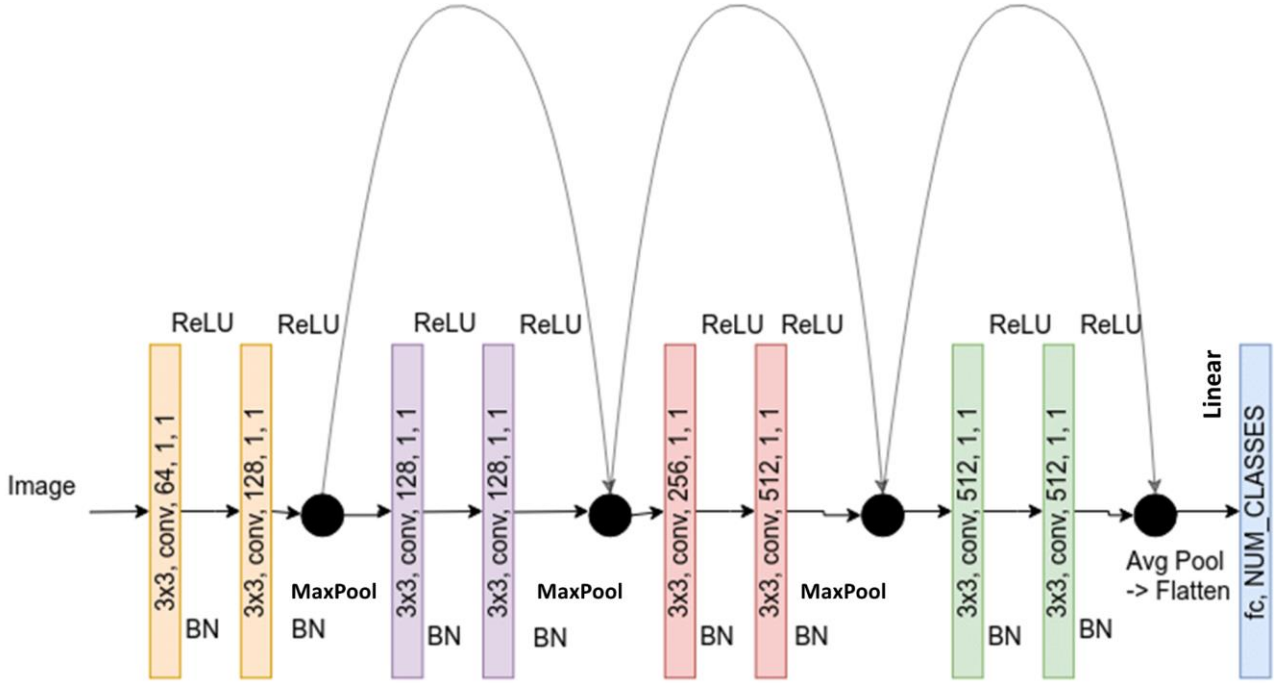
To address the dataset imbalance, we have merged all the data and split it into new train, validation and test directories containing 65%, 20% and 15% of the data respectively. The new dataset had 57,119 training samples, 17,566 validation samples, and 13,215 test samples. The training data was consistently spread out across the 38 classes as shown below.



All images were resized to 224x224 pixels, a standard input size for convolutional neural networks. Next, the images were normalized using the ImageNet mean and standard deviation values to ensure consistency in pixel value distribution.

Modeling

The model architecture chosen for this project is a Convolutional Neural Network (CNN) with residual connections. The architecture consists of four convolutional layers, each paired with batch normalization and ReLU activation for stability and improved convergence. MaxPooling layers were integrated to reduce spatial dimensions, and residual connections were added to overcome vanishing gradient issues. The final layer is a fully connected layer that outputs logits, which are converted to probabilities for each of the 38 classes during the loss computation. The loss function used here is Cross Entropy Loss, which incorporates the Softmax function internally.



We have considered alternatives such as plain CNNs and VGG networks, but our model is designed to address their limitations, such as vanishing gradient issues, ensuring more efficient training and better performance on deeper architectures. Although more complex models like Vision Transformers and DenseNet may achieve higher accuracy, their computational intensity makes them impractical for forecasting 38 disease classes in a resource-constrained environment.

Training

The Plant Friend model was executed using PyTorch and trained for 8 epochs. Custom data loaders were developed to manage the augmented dataset, ensuring efficient batch processing during training. The model was initialized with default weights, and the training loop incorporated forward passes, backpropagation, and gradient clipping to maintain stability. Early stopping and L2 Regularization was applied to prevent overfitting, and a learning rate scheduler dynamically adjusted the learning rate to optimize convergence. The hyperparameters for training was as follows:

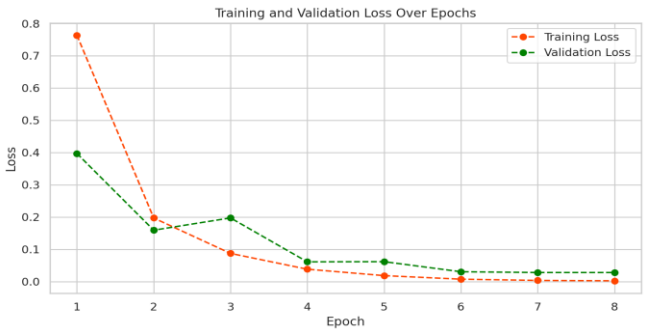
Batch size= 64 | Number of epochs = 8 | Max learning rate = 0.005

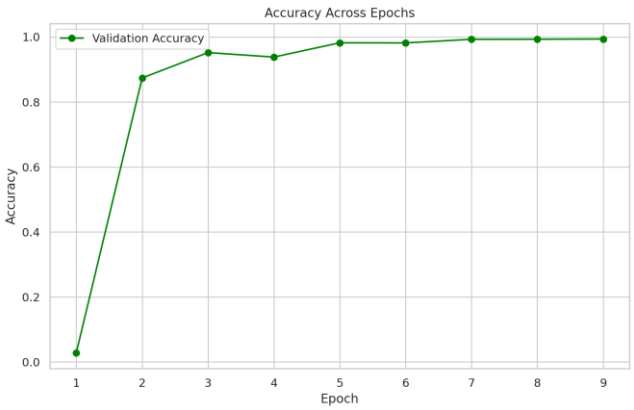
Gradient clip = 0.15 | Weight decay = 1e-4 | Optimizer function = SGD

Training on CPU and T4 GPU was extremely time consuming. Therefore, we had to shift to A100 GPU to reduce computation time drastically and enable iterative experimentation.

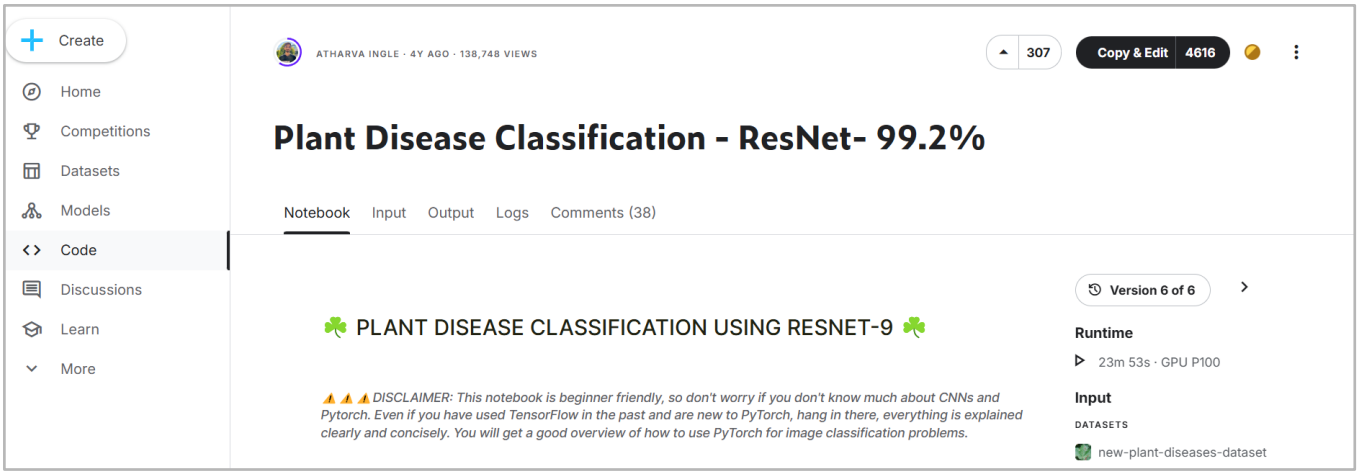
Validation

The CNN achieved a validation accuracy of 99.3% with a validation loss of 0.0282, indicating robust learning and convergence. This is evidenced further in the following graphs.





It is important to note that our model was able to beat a Gold Medal status solution on Kaggle which achieved validation accuracy of 99.2% using ResNet-9.

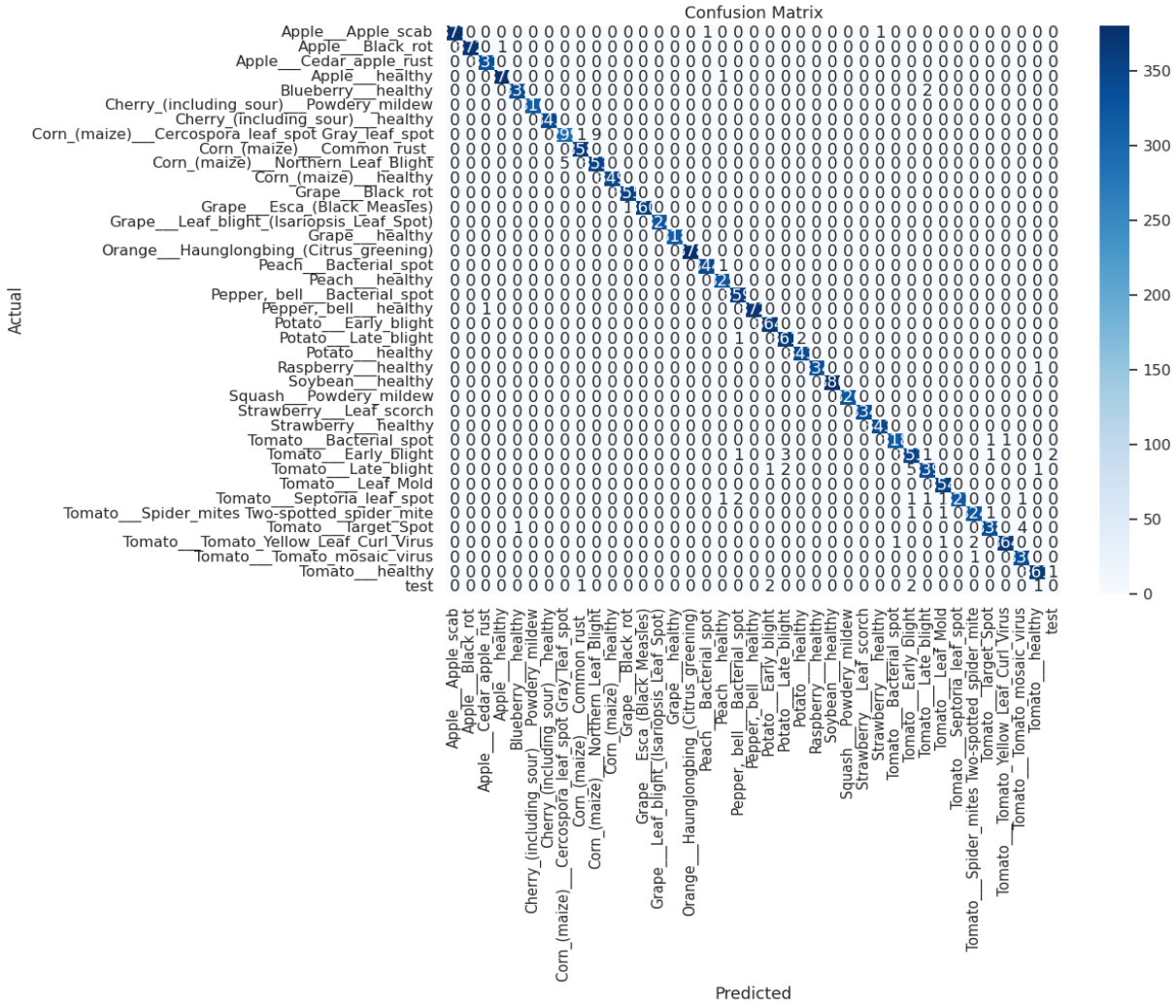


Differences	ResNet Gold Medal Status solution	Our Model
Batch Size	32	64
Optimizer	Adam	SGD
Max Learning Rate	0.01	0.005

Testing

On the test dataset, the model achieved an accuracy of 99.4%, demonstrating excellent generalization to unseen data. Precision, recall, and F1-scores for most classes were near or at 1.0, reflecting the model's reliability across all disease categories. For instance, "Apple___Apple Scab" achieved a precision of 1.000 and an F1-score of 0.997. A confusion matrix revealed minimal misclassifications, with high performance across all 38 classes.

The confusion matrix shows the model's classification performance across all 38 classes, showing minimal misclassifications. Classes with visual similarities, like "Apple___Cedar Rust" and "Apple___Black Rot," were the most challenging to distinguish, but even these achieved F1-scores above 0.95. This trend suggests that the model did not overfit the training data.

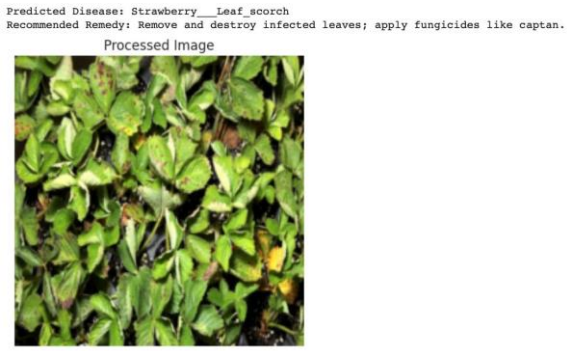


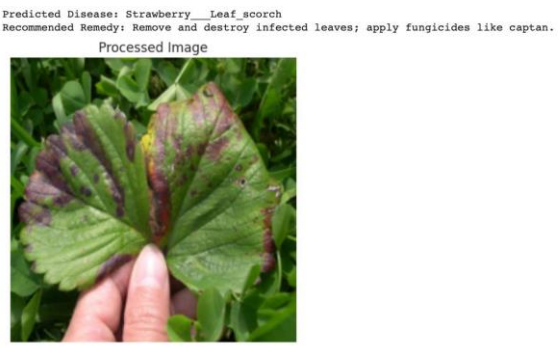
Remedy mapping

After ensuring that the model is robust, we focused on generating remedies for the different predicted diseases. Using data from the US department of Agriculture, a pandas dictionary was created mapping each of the 38 classes to the associated remedies and recommendations. The model was then tested out to check if it is predicting the right disease and generating the right remedy using test data. As shown below, the model was able to detect a healthy tomato leaf and suggest that no remedy is required.

Predicted Label: Tomato___healthy
Recommended Remedy: Plant is healthy, no remedy required.

Finally, the model was tested on unseen images from the internet. The model performs well for diseases like **Strawberry Leaf Scorch** as diverse images - multiple leaves and leaves held with hand generated accurate predictions and remedies.





The model faced challenges in accurately predicting diseases with visually similar symptoms. For instance, when an image of potato early blight was uploaded, the model incorrectly identified it as pepper bell bacterial spot. This misclassification likely occurred because potato and pepper bell leaves can appear similar, and their diseases share symptoms like discolored spots.

```
Predicted Disease: Pepper,_bell___Bacterial_spot
Recommended Remedy: Apply copper-based bactericides; use disease-free seeds.
```

To address this issue, it is crucial to include a broader and more diverse set of images during training, focusing on plant leaves and diseases with similar visual characteristics. This will help the model learn finer distinctions and improve its prediction accuracy.

Another key limitation is the model's inability to identify diseases outside the training classes, leading to misclassification of unseen conditions. To mitigate this, the model will be retrained periodically with diverse datasets, ensuring fairness and generalizability. Additionally, variations in image quality, lighting, and angles further impact accuracy, making the model less robust in real-world scenarios.

Deployment

The Plant Friend model is designed to be deployed as a mobile application with a subscription model, enabling real-time disease detection for farmers and agricultural organizations. The mobile application will allow farmers to upload leaf images and receive instant diagnoses and remedy suggestions once they subscribe. To ensure efficient deployment, the model will be exported using lightweight formats such as TensorFlow Lite, making it compatible with mobile and edge devices. Some potential issues with this approach are that mobile devices require sufficient computational power for inference, and maintaining cloud infrastructure may incur significant costs, especially during peak usage.

Farmers must also be informed that the tool serves as an assistive diagnostic aid and not as a definitive solution, especially given the potential for misclassification.

Ethical Considerations

Plant Friend risks amplifying inequalities if marginalized farmers lack access to the necessary technology or internet connectivity, potentially excluding them from its benefits. Additionally, frequent recommendations of chemical treatments could harm the environment through overuse and pollution, particularly if sustainable alternatives are overlooked. Addressing these challenges requires ensuring that the application is affordable and usable for farmers in low-resource regions through partnerships with agricultural organizations and local governments along with prioritizing eco-friendly treatment suggestions to minimize environmental impact.

Potential Risks

Some risks associated with this deployment plan include misclassifications that may lead to incorrect treatments, potentially harming crops. To mitigate this, the application will include confidence scores for predictions, encouraging farmers to consult experts for low-confidence cases. Additionally, feedback mechanisms will allow users to report errors, enabling continuous model improvement.

Over time, the model may degrade in accuracy due to emerging plant diseases or environmental changes; a retraining pipeline will address this by incorporating new data. Infrastructure downtime is another potential risk, which will be mitigated by using redundant servers and robust load-balancing techniques.

By addressing these challenges and risks, the Plant Friend project positions itself as a scalable and reliable solution that can be developed into a mobile app where farmers can subscribe and get solutions for their diseased plants.

Conclusion

The Plant Friend project successfully demonstrates the application of deep learning in agriculture, achieving high accuracy in plant disease detection. With validation accuracy of 99.3% and test accuracy of 99.4%, the model is both reliable and scalable. Future improvements include expanding the dataset to include more plant species, including more diverse images for diseases and leaves that look similar, and deploying the solution for low-resource environments. By integrating advanced machine learning techniques with real-world agricultural needs, Plant Friend has the potential to function as a viable revenue-earning platform, making a meaningful impact on global food security.