Leveraging Custom Scalable CNN Framework for Efficient Plant Disease Classification

Prasenjit Datta[1], Subhrajit Ghosh[1], Aditri Chaudhuri[1], Soumyadip Dutta[1], Debangshu Roy[1], Ishanee Ghosh[2], Supriya Sarkar[3], Bidhan Barai[4,*], and Pawan Kumar Singh[5,6]

[1] Department of Computer Science and Engineering (AI-ML), Techno Main Salt Lake, EM-4/1, Sector-V, Salt Lake, Kolkata-700091, West Bengal, India
pr.datta.2021@gmail.com, subhrajitghosh35@gmail.com, aditric@gmail.com, soumyadipduttarajofficial@gmail.com, roy.debangshu2023@gmail.com

[2] Department of Computer Science and Engineering, Techno Main Salt Lake, EM-4/1, Sector-V, Salt Lake, Kolkata-700091, West Bengal, India
ig.ghosh2019@gmail.com

[3] Department of Computer Science and Engineering (AI-ML), Techno Main Salt Lake, EM-4/1, Sector-V, Salt Lake, Kolkata-700091, West Bengal, India
supriyasarkar2501@gmail.com

[4] Department of Computer Science and Engineering, SRM University AP, Mangalagiri Mandal, Neeru Konda, Amaravati, Andhra Pradesh 522502
bidhan.ju.cse@gmail.com

[5] Department of Information Technology, Jadavpur University, Jadavpur University Second Campus, Plot No. 8, Salt Lake Bypass, LB Block, Sector III, Salt Lake City, Kolkata, Pin: 700106, West Bengal, India

[6] Shinawatra University, 99, Moo 10, Bang Toei, Sam Khok, Pathum Thani, Thailand, 12160
pawansingh.ju@gmail.com

Abstract. Plant diseases pose a significant threat to global agricultural productivity, requiring efficient and scalable solutions for early detection and management. This paper presents a deep learning-based system for the detection of plant diseases, utilizing computational neural networks (CNNs) to classify and diagnose plant health problems in five different crops: corn, potato, rice, tea, and tomato. The system incorporates a dual-model approach: fine-tuning the pre-trained ResNet50 model and designing a custom CNN architecture. Both models are optimized for robust feature extraction and high-accuracy classification, utilizing techniques such as data augmentation, preprocessing, and hyperparameter tuning. A data set comprising over 20,000 labeled images is processed through resizing, normalization, and augmentation to improve model generalization and reduce overfitting. The standard pre-trained ResNet50 model, fine-tuned with a modified Softmax layer, achieves superior classification accuracy due to its advanced residual learning framework. Meanwhile, the custom CNN architecture balances computational efficiency and performance, making it suitable for deployment on resource-constrained systems. The system is evaluated using metrics such as precision, precision, recall, and F1 score, demonstrating exceptional

performance, particularly for crops such as potatoes, rice, and tomatoes, where classification precision exceeds 97%. This work underscores the transformative potential of deep learning in agricultural diagnostics, offering accurate and real-time plant disease identification to enhance crop management and reduce losses.

Keywords: Machine learning · Plant disease detection · Image processing · ResNet50 · Convolutional neural network

1 Introduction

Agriculture is the backbone of the global economy, and crop diseases pose a pressing challenge, especially in developing regions where access to expert assistance is limited. Crop health is a critical determinant of productivity, and the prevalence of diseases significantly affects food security, leading to economic losses and reduced yields. Conventional disease identification techniques, which depend on skilled human inspection, are expensive, time-consuming, and frequently unavailable to small-scale farmers. Recent advances in artificial intelligence (AI) and computer vision (CV) have provided a powerful alternative. This paper focuses on using machine learning (ML) models for the detection of crop diseases using leaf images, enabling early diagnosis and timely treatment. Our system employs a hierarchical framework: first, identifying whether a plant is healthy or infected, and then, predicting the specific disease, if infected. Scalability and increased accuracy for various types of crops are guaranteed by this modular approach. The deployed model is accessible through a web-based interface, providing farmers and agricultural experts with a user-friendly tool for real-time disease diagnosis.

The primary objective of this research is to design a robust and scalable deep learning framework to detect crop diseases accurately. The system initially determines whether a plant is healthy or diseased using a hierarchical approach, and if it is, it identifies the particular illness. The framework aims to ensure adaptability across diverse crop types while maintaining high accuracy and usability. Furthermore, the study aims to develop a user-friendly interface to make advanced crop disease diagnosis accessible to farmers and agricultural experts worldwide. The **source codes** related to this work can be found at this link [1].

2 Literature Review

Ramesh et al. [15] proposed a multi-step method was suggested for identifying whether leaves are diseased or healthy, utilizing feature extraction techniques such as *HoG*, *Hu moments*, *Haralick texture features*, and *color histograms*. The Random Forest Classifier showed exceptional accuracy i.e. 70.14%, especially with smaller datasets, surpassing other methods.

Harakannanavar et al. [8] developed a hybrid model to identify tomato leaf diseases that combines *CNN*, *DWT*, *PCA*, and *Grey Level Co-occurrence Matrix (GLCM)*. The model's accuracy was evaluated on tomato samples with disorders, achieving 88% with SVM, 97% with K-NN, and 99.6% with CNN.

Chohan et al. [6] utilized CNNs with *TensorFlow* and *Keras* for plant disease detection using the PlantVillage dataset. Their model achieved 98.3% accuracy, with *Tomato Mosaic Virus* showing near-perfect results, highlighting CNNs' effectiveness in plant disease detection.

Suresh et al. [19] developed an Android app using *TFLite* and *CNNs* for detecting plant diseases, linking users to e-commerce platforms for pesticide purchases. The system outperformed with accuracy 90.723% classified by using ANN and nearest neighbour algorithms.

Chopda et al. [7] used feature selection algorithms to predict pest incidence on cotton plants based on climatic parameters. *Recursive Feature Elimination (RFE)* combined with *clustering* helped predict pest effects using meteorological data.

Shelar et al. [17] developed a neural network model for leaf disease detection using *TensorFlow Lite*. The model, based on *VGG-19* architecture, achieved 95.6% accuracy for strawberry and potato diseases, demonstrating deep learning's potential for mobile-based agricultural diagnostics.

Mohanty et al. [12] proposed a deep CNN model, based on *AlexNet* and *GoogleNet* architecture for accurately identifying 14 crop species and 26 diseases. The model achieved an overall accuracy of 99.35% (while using GoogleNet architecture), with a 13.82% improvement in accuracy over traditional AlexNet approaches.

Ramcharan et al. [14] created a model for plant disease prediction using *Inception v3 CNN* model and with the help of techniques like segmentation and feature extraction. Neural network-based topologies achieved overall accuracy of 93% (for Leaflet Cassava dataset) and 91% (for the original Cassava Dataset) in classifying plant diseases , both using the SVM architecture on top of pretrained Inception v3 model. If considered for individual class it showed an 98% accuracy for *Brown Leaf Spot (BLS)* and *Cassava Brown Streak Disease (CBSD)*.

Panchal et al. [13] employed a deep learning approach to classify healthy and diseased apple leaves using the PlantVillage dataset. Transfer learning with *VGG16* achieved the best accuracy of 93.5% , outperforming ResNet and other architectures.

Khirade and Patil [9] used digital *image processing* and *BPNN* for plant dis-

ease detection. Their methods included thresholding, edge detection, and texture analysis to classify plant diseases, showing the potential of neural networks for plant disease identification.

Moghadam et al. [10] applied *hyperspectral imaging* in plant disease diagnosis, using *VNIR* and *SWIR spectra* and achieved accuracy of 91.5%. Their approach RBF SVM achieved 93.6% accuracy, though the need for specialized equipment limits scalability.

Sharath et al. [16] developed a method to detect *bacterial blight* in *pomegranates* using color, texture, and edge-based features. Their system successfully predicted infection severity.

Shrestha et al. [18] used *CNNs* for plant disease classification, achieving 88.8% accuracy across 12 plant diseases. However, the model's F1 score of 0.12 indicated issues with false negatives.

Badage [3] study designs an automatic system that detects plant illnesses at an early stage through remote sensing images and machine learning methods. The system allows agriculturists to check big farm areas and spot wheat and cotton plant diseases (for instance rusts and mildew) through *Canny's edge detection* and machine learning analysis. Timely action becomes possible when diseases are found early to prevent crop loss and increase output.

Mohameth et al. [11] investigated transfer learning using *VGG16*, *GoogleNet*, and *ResNet* for plant disease detection. VGG16 performed best with 97.92% accuracy, highlighting the importance of model selection for optimal performance.

Bhise et al. [5] developed a mobile app using *TensorFlow Lite* and *CNNs* to detect plant diseases. The system showed high accuracy and demonstrated the effectiveness of lightweight models for real-time disease detection on mobile devices.

Alagumariappan et al. [2] built a decision support system for plant disease detection using *Raspberry Pi*, *Hu moments*, and *Haralick texture features*. The system outperformed SVM with ELM, showing real-time classification efficiency.

Too et al. [20] utilized CNN architectures like *VGG16*, *Inception V4*, *ResNet*, and *DenseNet* for plant disease classification and achieved the accuracy of 99.75%. The study demonstrated optimized deep learning techniques for scalable, accurate plant disease management solutions.

Barbedo [4] studied plant disease classification using expanded datasets and transfer learning with *GoogLeNet* and achieved outperforming accuracy using individual lesions and spots was 94%. The approach improved classification ac-

curacy, highlighting the potential of data augmentation and transfer learning for robust disease detection.

The works cited in the literature review serve as a basis for our investigation by demonstrating several methods for identifying plant diseases. Mohanty et al. [12] demonstrated the potential of CNN-based models, inspiring our use of ResNet50. Similarly, Harakannanavar et al. [8] and Chohan et al. [6] emphasized data augmentation and transfer learning to enhance model robustness, approaches we adopted to improve performance. Studies by Chopda et al. [7] and Suresh et al. [19] explored a way of integrating environmental data, aligning with our aim to develop a scalable system. Moreover, Shelar et al. [17] addressed challenges in achieving high accuracy across multiple crops, motivating us to test our model on diverse data sets. By synthesizing insights from these works, our study focuses on developing a robust system and advancing the field of plant disease detection.

3 Proposed Methodology

The proposed methodology integrates ResNet50, a pre-trained deep learning model renowned for its residual learning capabilities, to leverage its advanced feature extraction and classification performance. ResNet50's architecture is fine-tuned on our crop disease dataset, enabling it to learn domain-specific patterns effectively. Its residual block helps mitigating the *vanishing gradient* issues during training, ensuring robust performance for five crop classes. The model processes input images preprocessed to a resolution of 128x128, yielding high-accuracy predictions suitable for complex agricultural applications. We capitalize on state-of-the-art transfer learning techniques to enhance the reliability of disease detection utilizing *ResNet50*.
Alongside ResNet50, a custom Convolutional Neural Network (CNN) model is designed which includes optimized convolutional, pooling, and fully connected layers that extract and classify critical features unique to the images of plant leaves in our data set. It focuses on computational efficiency without compromising classification accuracy, making it suitable for the deployment of resource-constrained systems. Input images undergo pre-processing similar to ResNet50 for consistency, and the model is trained from scratch to ensure adaptability to the unique characteristics of the dataset. This dual-model approach enables us to evaluate and combine the strengths of both architectures for an effective and scalable crop disease detection system.

Table 1 lists the type, size of the output layer, and number of parameters for each layer in the CNN architecture. It includes *Conv2D*, *MaxPooling2D*, *Dropout*, *Flatten*, and *Dense* layers, highlighting hierarchical design and parameters of the network.

Table 1: Detailed architectural information of the layers, output shape, and number of parameters present in the proposed pre-trained CNN model

Layer (type)	**Output Shape**	**Param #**
conv2d_10 (Conv2D)	(None, 128, 128, 32)	896
conv2d_11 (Conv2D)	(None, 126, 126, 32)	9248
max_pooling2d_5 (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_12 (Conv2D)	(None, 63, 63, 64)	18496
conv2d_13 (Conv2D)	(None, 61, 61, 64)	36928
max_pooling2d_6 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_14 (Conv2D)	(None, 30, 30, 128)	73856
conv2d_15 (Conv2D)	(None, 28, 28, 128)	147584
max_pooling2d_7 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_16 (Conv2D)	(None, 14, 14, 256)	295168
conv2d_17 (Conv2D)	(None, 12, 12, 256)	590080
max_pooling2d_8 (MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_18 (Conv2D)	(None, 6, 6, 512)	1180160
conv2d_19 (Conv2D)	(None, 4, 4, 512)	2359808
max_pooling2d_9 (MaxPooling2D)	(None, 2, 2, 512)	0
dropout_2 (Dropout)	(None, 2, 2, 512)	0
flatten_1 (Flatten)	(None, 2048)	0
dense_2 (Dense)	(None, 1500)	3073500
dropout_3 (Dropout)	(None, 1500)	0
dense_3 (Dense)	(None, 39)	58539

3.1 Dataset Used

Five different plant datasets from Kaggle have been used to train our model, as mentioned in Table 2. The table shows all the disease categories of each dataset and the total number of images present.

Table 2: Information of the datasets used

Specimen	**Categories**	**Total images**
Corn	Blight, Common Rust, Gray Leaf Spot, Healthy	4188
Potato	Early Blight, Late Blight, Healthy	2152

Specimen	Categories	Total images
Rice	Bacterial Leaf Blight, Brown Spot, Healthy Rice Leaf, Leaf Blast, Leaf Scald, Sheath Blight	3829
Tea	Anthracnose, Algal Leaf, Bird Eye Spot, Brown Blight, Gray Light, Healthy, Red Leaf Spot, White Spot	885
Tomato	Tomato Late Blight, Tomato Healthy, Tomato Early Blight, Tomato Septoria Leaf Spot, Tomato Yellow Leaf Curl Virus, Tomato Bacterial Spot, Tomato Target Spot, Tomato Mosaic Virus, Tomato Leaf Mold, Tomato Spider Mites Two-spotted Spider Mite	10000

3.2 Data Splitting

Dataset is partitioned into three subsets: **Training Dataset** (used for learning during model training), **Validation Dataset** (evaluated during training to tune hyperparameters and assess intermediate performance), and **Test Dataset** (assessed post-training to evaluate model generalization on unseen data).

3.3 Data Optimization and Augmentation

Efficient data handling during training is achieved through

1. **Caching:** Stores the dataset in memory to eliminate repeated disk access, accelerating training by enabling rapid data retrieval.
2. **Shuffling:** Randomizes data order to prevent unintended learning patterns. A buffer size of 1,000 ensures sufficient randomness.
3. **Prefetching:** Overlaps data preprocessing with model computation by preloading batches while the model processes current data. The AUTOTUNE setting optimizes this process based on system resources.

These enhancements maximize throughput, improve model generalization, and leverage computational resources effectively. Normalizing image pixel values to the range [0, 1] is integrated into the data pipeline for consistent preprocessing during training and inference.
Data augmentation increases dataset diversity and improves model robustness by simulating real-world variations.

3.4 Model Building and Training

Algorithm 1 demonstrates the architecture and implementation details of the pre-trained ResNet50 model whereas Algorithm 2 demonstrates the architecture and implementation details of the proposed custom CNN model.

Algorithm 1 Plant Classification (Using Pre-trained ResNet50 Model)

Input: Crop images (Rice, Corn, Tea, Potato, Tomato)

Step 1: Load and Pre-process Data
Procedure:
- Load crop images dataset (Rice, Corn, Tea, Potato, Tomato)
- For each image
  - Resize images to 128 x 128 pixels
  - Normalize pixel values to range [0, 1]
  - Convert images to tensors for PyTorch

Step 2: Data Augmentation
Procedure:
Apply random augmentations
- Random rotation (0 to 30 degrees)
- Random horizontal and vertical flips
- Random zooming and cropping

Step 3: Model Design
Procedure:
- Initialize ResNet50 pre-trained on ImageNet dataset
- Replace the final fully connected layer with 5 units (for 5 plant types)
- Set the activation function of the final layer to softmax
- Use **Cross-Entropy Loss** for multi-class classification

Step 4: Compile and Train the Model
Procedure:
- Set **Optimizer** to Adam
- Set **Loss Function** to Cross-Entropy Loss
- Set **Metrics** to Accuracy
- Train the model using training data

Step 5: Model Evaluation
Procedure:
- Evaluate the model on the test dataset
- Calculate accuracy, precision, recall, and F1-score
- Generate a confusion matrix to analyze misclassification images

Step 6: Fine-Tuning
Procedure:
- Unfreeze the last few layers of ResNet50
- Tune hyperparameters (learning rate, batch size)
- Retrain the model to improve accuracy

Algorithm 2 Disease Detection for Each Crop (Using Custom CNN Model in TensorFlow)

Step 1: Load and Pre-process Data
Procedure:

- Load images for each crop: Rice, Corn, Tea, Potato, Tomato (healthy vs. infected)
- Resize images to 128x128 pixels
- Normalize pixel values to range [0, 1]
- Convert images to tensors for TensorFlow

Step 2: Data Augmentation
Procedure:
Apply random augmentations
- Random rotation (0 to 20 degrees)
- Random horizontal and vertical flips
- Random brightness adjustment
- Random zooming

Step 3: Split Dataset
Procedure:
Split the dataset into
- **Training Set:** 70%
- **Validation Set:** 15%
- **Test Set:** 15%

Step 4: Model Design
Procedure:
Design a custom CNN with layers
- **Convolutional Layers:** 32 filters, 64 filters, etc
- **MaxPooling Layers:** For down-sampling
- **Dropout Layer:** To prevent overfitting
- **Fully Connected Layers:** Dense layer with 128 units
- **Output Layer:** Use a single neuron with a sigmoid activation (binary classification: healthy/infected)
- Use **Binary Cross-Entropy Loss**
- **Total Parameters:** 7,844,263

Step 5: Compile and Train the Model
Procedure:
- Set **Optimizer** to Adam
- Set **Loss Function** to Binary Cross-Entropy
- Set **Metrics** to Accuracy
- Train the model on the training dataset, monitoring performance on the validation set

Step 6: Model Evaluation
Procedure:
- Evaluate the model on the test dataset
- Calculate accuracy, precision, recall, and F1-score
- Generate a confusion matrix to analyse misclassifications

Step 7: Fine-Tuning
Procedure:
- Perform hyperparameter tuning (learning rate, batch size)
- Retrain the model to improve performance

3.5 Error Analysis

To understand what went wrong, we performed an error analysis of the specific cases where the models failed after the system was evaluated. We tried to identify all the misclassifications and the root causes.

1. **Misclassification Insights:** During testing, it was noticed that the plant classification model (ResNet50) sometimes misclassified healthy leaves with different visual patterns as diseased, especially when the background noise was high. Similarly, the custom CNN model for disease detection struggled with cases where symptoms like discoloration or lesions overlapped across multiple diseases (e.g., Late Blight and Early Blight in tomatoes).
2. **Dataset Challenges:** A significant source of error was due to inconsistencies in the dataset. Images with poor lighting, low resolution, or significant occlusions often led to incorrect predictions. This highlights the need for a more robust dataset preprocessing pipeline to handle noisy inputs effectively.
3. **Class Imbalance:** Some diseases were underrepresented in the dataset, and thus there were higher false negative rates for those classes. For example, the classes of diseases with fewer training samples, such as Tomato Mosaic Virus, had a higher misclassification rate than well-represented classes like Tomato Late Blight.

3.6 Scalability related challenges

1. **Data diversity and quality:** Different agricultural environments differ widely in terms of soil, climate, crops, and agricultural practices. Data collection, diversity, and labeling are highly labor-intensive and costly.
2. **Infrastructure constraints:** Reputation of many rural regions, especially in underdeveloped countries, is that of having intolerable connectivity to enable real-time data transmission. Deployment and maintenance of these IoT devices and sensors could be expensive for smallholder farmers.
3. **Scalability computational efficiency:** Maintaining a significant level of performance for large-scale adoption is demanded. Real-time diseased detection relies on effective processing of streams of data and optimized ML models.
4. **Adoption and training:** Low digital literacy makes it difficult for the farmers to operate the technology. A user-friendly interface, multilingual support, and training will be necessary for user uptake.
5. **Regulatory and environmental issues:** Different agricultural policies, data privacy laws, and pesticide legislation have therefore to be considered based on the regions. Given that the system is supposed to lead to an increase in fertilizer or chemical use, sound eco-practices must be built in.

3.7 Deployment

Fig. 1 shows the schematic of the overall architecture for plant disease classification. The proposed model has been deployed using **FastAPI** for API interactions, **Docker** for containerization, **Kubernetes** for orchestration, and **AWS**

for secure and scalable cloud hosting, ensuring reliability and efficiency.

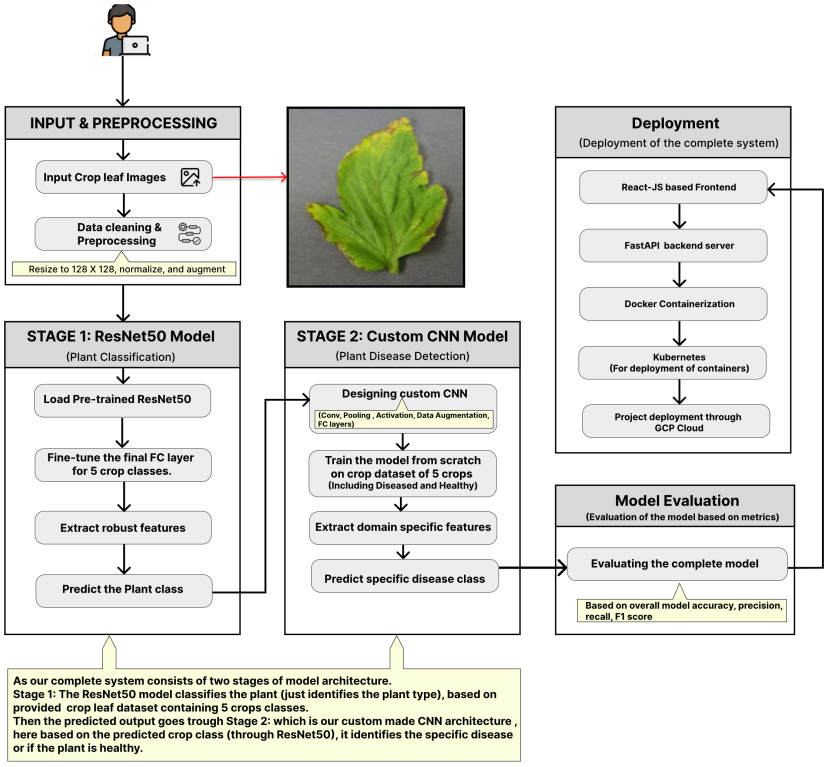


Fig. 1: Overall architecture of our proposed model for plant disease classification.

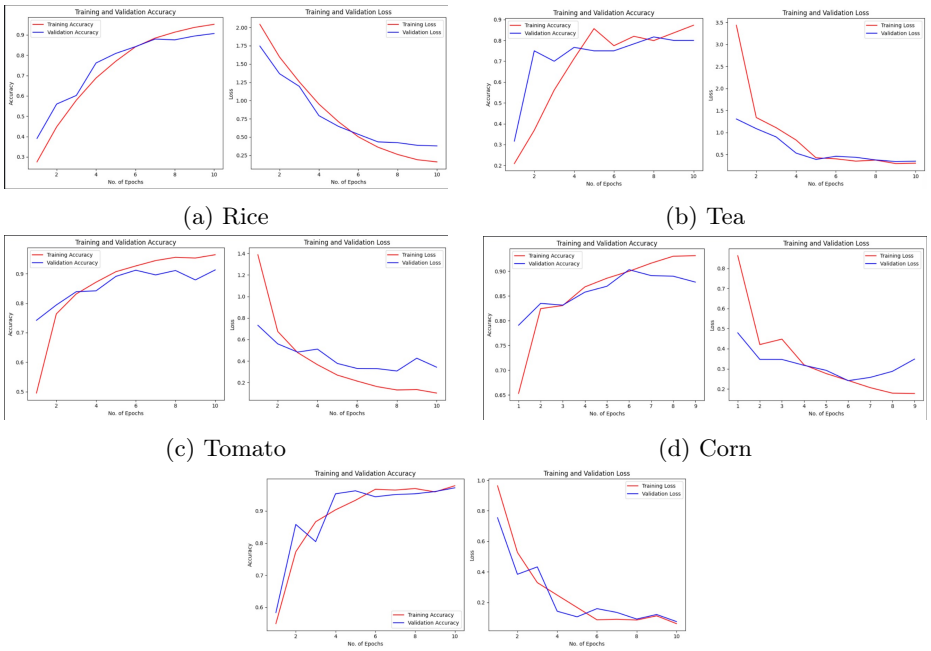
4 Results and Discussion

Fig. 2 illustrates the **training and validation accuracy** alongside **training and validation loss** for the five crops analyzed in this study: rice, tea, tomato, corn, and potato. Each subplot represents the respective crop's performance during the training process.

– **Accuracy Graphs:** The training accuracy (in red) and validation accuracy (in blue) demonstrate a consistent upward trend for all crops, indicating improved model learning over epochs. Minor fluctuations in validation accuracy reflect the complexity of the dataset and the model's generalization capabilities.

– **Loss Graphs:** Similarly, the training loss (in red) and validation loss (in blue) exhibit a consistent downward trajectory, highlighting the effective minimization of the error during training. Any divergence between the training and validation curves is carefully analyzed to detect potential overfitting.

For crops like potato, tomato, and rice, the validation accuracy surpasses 97% after convergence, showcasing the model's robustness. Meanwhile, for crops such as tea and corn, minor variances in validation performance suggest scope for further tuning.



(a) Rice (b) Tea

(c) Tomato (d) Corn

(e) Potato

Fig. 2: Training and validation accuracy curve and loss curve given by the proposed pre-trained CNN ResNet50 model.

Table 3 shows performance metrics (in terms of training accuracy, validation accuracy, Precision, and Recall) for all the five different plant datasets.

1. **Corn:** The Corn model exhibited strong performance with 93.17% training accuracy and 90.32% validation accuracy, demonstrating effective generalization. Its precision of 90.56% showed proficiency in correctly identifying Corn samples, while the 87.69% recall indicated a high capacity for capturing relevant Corn data.
2. **Potato:** The Potato crop model displayed exceptional results, achieving 99.13% training accuracy and 97.21% validation accuracy, signifying almost

flawless performance across both datasets. With 91.14% precision, it accurately predicted Potato instances, though the slightly lower 85.00% recall suggests potential for enhancement in identifying all Potato samples.

3. **Rice:** For Rice, the model attained 97.86% training accuracy and 90.68% validation accuracy. Its 89.50% precision reflected effective avoidance of false positives, while the higher 88.32% recall demonstrated good sensitivity in detecting Rice instances.
4. **Tea:** The Tea model reached 93.85% training accuracy but faced challenges with a lower 80.00% validation accuracy, indicating possible overfitting or generalization issues. Its 84.00% precision suggested moderate prediction reliability, while the 82.21% recall was somewhat behind other crops in identifying all relevant cases.
5. **Tomato:** The Tomato model performed admirably, achieving 97.14% training accuracy and 91.10% validation accuracy. Its 89.40% precision ensured accurate Tomato identification, complemented by 86.00% recall, indicating robust sensitivity in detecting Tomato instances.

Table 3: Performance metrics given by the proposed pre-trained CNN ResNet50 model.

Plant	**Training accuracy**	**Validation accuracy**	**Precision**	**Recall**
Corn	93.17%	90.32%	90.56%	87.69%
Potato	99.13%	97.21%	91.14%	85.00%
Rice	97.86%	90.68%	89.50%	88.32%
Tea	93.85%	80.00%	84.00%	82.21%
Tomato	97.14%	91.10%	89.40%	86.00%

The practical implications of the proposed system are noteworthy

1. **Cost Savings:** The system eliminates the need for labor-intensive manual disease scouting, reducing associated labor costs while minimizing errors.
2. **Time Efficiency:** Real-time detection ensures early identification of crop diseases, thus allowing farmers to take timely preventive measures, thereby saving critical response time.
3. **Resource Optimization:** By leveraging pre-trained CNN model, the system achieves high performance without requiring extensive computational resources, making it cost-effective and accessible.
4. **Scalability and Usability:** The lightweight design and its compatibility with mobile and IoT-based platforms will make the system deployable in remote resource-constrained regions, thereby increasing its accessibility among farmers.
5. **Improved Productivity:** With accurate disease diagnosis, the model can help reduce crop losses and improve yields, leading to higher agricultural productivity.

6. **Sustainability:** With its ability to provide precise and timely diagnostics, the system supports sustainable farming practices by reducing pesticide misuse and improving resource management.

5 Conclusion

This research showcases an effective method for identifying plant diseases using ResNet50, a pre-trained convolutional neural network (CNN) model. The system processes datasets from corn, potato, rice, tea, and tomato plants, employing preprocessing, data augmentation, and optimization strategies to enhance training input quality and computational efficiency. The model exhibited impressive performance, with training accuracies surpassing 97% for potato, rice, and tomato crops, and validation accuracies reaching 97.21%, 91.10%, and 90.68% for potato, tomato, and rice, respectively. The model's reliability is further evidenced by precision and recall metrics, with potato achieving the highest precision at 91.14%. The CNN architecture, comprising over 180,000 trainable parameters, effectively captured intricate patterns in image data, successfully differentiating between healthy and diseased plants while maintaining a balance between performance and scalability. These findings highlight the potential of deep learning in addressing agricultural challenges by providing accurate, real-time disease identification, ultimately contributing to improved crop productivity.

5.1 Key Observations

The present work highlighted the importance of combining robust machine learning techniques with scalable deployment strategies. Key findings from the development and testing phases are summarized below.

- Data augmentation significantly improved model generalization and reduced overfitting.
- A multi-model approach enhanced prediction accuracy for plant health and disease detection
- Optimized hyperparameters, such as a learning rate of 1e-3 and batch size of 32, balanced efficiency and performance.
- Deployment using FastAPI, Docker, and Kubernetes ensured scalability and reliability on AWS infrastructure.

5.2 Future Works

Future work will focus on enhancing model accuracy by incorporating more diverse datasets and advanced architectures. Additional features like real-time monitoring, multi-language support, and integration with IoT devices for automated alerts will be explored to improve usability and scalability.
Integrating IoT devices and real-time monitoring presents both broad opportunities and significant challenges. Such advancements would allow for automated

data collection from sensors deployed in the field, which would continue to update information regarding crop health, environmental conditions, and disease progression. It would allow proactive interventions, thereby reducing response time and minimizing the potential damage to crops. However, implementation will have to surmount severe barriers: expensive and sometimes very complex IoT-deployment infrastructure; access and coverage across vast and disparate rural and regional geographies; handling voluminous real-time data that's exponentially larger and needs more space in servers for further processing and mining; also interoperability with various devices without having some sort of standard protocol among those IoT-devices, the time difference when connecting the data real-time in such systems. Despite these drawbacks, such findings hold much hope to transform the area of precision agriculture and facilitate scalable solutions against disease.

References

1. Cnn model source code, https://github.com/D-roy-2003/Leveraging-Custom-Scalable-CNN-Framework-for-Efficient-Plant-Disease-Classification.git
2. Alagumariappan, P., Dewan, N.J., Muthukrishnan, G.N., Raju, B.K.B., Bilal, R.A.A., Sankaran, V.: Intelligent plant disease identification system using machine learning. Engineering Proceedings **2**(1), 49 (2020)
3. Badage, A.: Crop disease detection using machine learning: Indian agriculture. Int. Res. J. Eng. Technol **5**(9), 866–869 (2018)
4. Barbedo, J.G.A.: Plant disease identification from individual lesions and spots using deep learning. Biosystems engineering **180**, 96–107 (2019)
5. Bhise, N., Kathet, S., Jaiswar, S., Adgaonkar, A.: Plant disease detection using machine learning. International Research Journal of Engineering and Technology (IRJET) **7**(7), 2924–2929 (2020)
6. Chohan, M., Khan, A., Chohan, R., Katpar, S.H., Mahar, M.S., et al.: Plant disease detection using deep learning. International Journal of Recent Technology and Engineering **9**(1), 909–914 (2020)
7. Chopda, J., Raveshiya, H., Nakum, S., Nakrani, V.: Cotton crop disease detection using decision tree classifier. In: 2018 International Conference on Smart City and Emerging Technology (ICSCET). pp. 1–5. IEEE (2018)
8. Harakannanavar, S.S., Rudagi, J.M., Puranikmath, V.I., Siddiqua, A., Pramodhini, R.: Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings **3**(1), 305–310 (2022)
9. Khirade, S.D., Patil, A.B.: Plant disease detection using image processing. In: 2015 International conference on computing communication control and automation. pp. 768–771. IEEE (2015)
10. Moghadam, P., Ward, D., Goan, E., Jayawardena, S., Sikka, P., Hernandez, E.: Plant disease detection using hyperspectral imaging. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA). pp. 1–8. IEEE (2017)
11. Mohameth, F., Bingcai, C., Sada, K.A.: Plant disease detection with deep learning and feature extraction using plant village. Journal of Computer and Communications **8**(6), 10–22 (2020)

12. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Frontiers in plant science **7**, 1419 (2016)
13. Panchal, A.V., Patel, S.C., Bagyalakshmi, K., Kumar, P., Khan, I.R., Soni, M.: Image-based plant diseases detection using deep learning. Materials Today: Proceedings **80**, 3500–3506 (2023)
14. Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., Hughes, D.P.: Deep learning for image-based cassava disease detection. Frontiers in plant science **8**, 1852 (2017)
15. Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., Vinod, P., et al.: Plant disease detection using machine learning. In: 2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C). pp. 41–45. IEEE (2018)
16. Sharath, D., Kumar, S.A., Rohan, M., Prathap, C., et al.: Image based plant disease detection in pomegranate plant for bacterial blight. In: 2019 international conference on communication and signal processing (ICCSP). pp. 0645–0649. IEEE (2019)
17. Shelar, N., Shinde, S., Sawant, S., Dhumal, S., Fakir, K.: Plant disease detection using cnn. In: ITM Web of Conferences. vol. 44, p. 03049. EDP Sciences (2022)
18. Shrestha, G., Das, M., Dey, N., et al.: Plant disease detection using cnn. In: 2020 IEEE applied signal processing conference (ASPCON). pp. 109–113. IEEE (2020)
19. Suresh, V., Gopinath, D., Hemavarthini, M., Jayanthan, K., Krishnan, M.: Plant disease detection using image processing. International Journal of Engineering Research & Technology (IJERT) **9**, 78–82 (2020)
20. Too, E.C., Yujian, L., Njuki, S., Yingchun, L.: A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture **161**, 272–279 (2019)