International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Plant Leaf Disease Prediction Using Cnn

Sanjaykumar R[1], Mrs. V. Latha Sivasankari[2]

[1]Final MCA, Department of Computer Applications Hindusthan College of Arts and Science Coimbatore–28
[2]Assistant Professor, Department of Computer Applications Hindusthan College of Arts and Science Coimbatore–28

Abstract:
Plant diseases pose a significant threat to the livelihoods of smallholder farmers, impacting both income and food security. However, the widespread adoption of smartphones and advancements in computer vision have opened up new possibilities for image-based disease diagnosis in agriculture. Convolutional Neural Networks (CNNs) are at the forefront of image recognition technologies and have demonstrated their potential in providing accurate and timely disease identification.
This study explores the performance of a pre-trained ResNet34 model for detecting plant diseases. The model was trained and validated using a dataset comprising 8,685 leaf images, captured under controlled conditions, and is designed to identify seven distinct plant diseases as well as differentiate healthy leaf tissue. The developed system is deployed as a web-based application, enabling users to access its diagnostic capabilities conveniently.
The results indicate that the model achieves a validation accuracy of 97.2% and an F1 score exceeding 96.5%. These findings highlight the effectiveness of CNNs in plant disease classification and their potential to provide AI-driven solutions for smallholder farmers. This approach demonstrates a scalable and practical method for enhancing agricultural productivity and ensuring food security.

Keywords: Deep learning, plant disease identification, CNN, ResNet34, smallholder farmers, image classification.

1. INTRODUCTION:

Crop production plays a vital role in India’s economy and food security, necessitating advancements in agricultural technology to boost productivity. Traditionally, plant disease detection relies on manual observation, which is time-intensive, less accurate, and limited in scope. Automated techniques, particularly those using image processing, provide a faster, more accurate alternative by analyzing images to identify plant health conditions. This study focuses on developing a leaf recognition system leveraging image characteristics to classify plant diseases effectively. Using tools such as OpenCV, TensorFlow, and NumPy, the system processes images through resizing and labeling, enabling disease classification with minimal human intervention.
The classification process employs a Convolutional Neural Network (CNN), designed with multiple layers including convolutional, pooling, and regression layers. A key parameter, the learning rate (set at 1e-3), regulates the model's learning speed during training. The model is trained on pre-existing datasets of plant leaf images, resized to 50x50 pixels, and achieves high accuracy by reducing overfitting through a comprehensive training dataset. Once trained, the model identifies whether a leaf is healthy or diseased,

further categorizing infections caused by fungi, bacteria, or viruses. The system also provides recommendations for managing identified diseases such as powdery mildew, black spots, and downy mildew.

This approach overcomes the challenges of limited datasets and low detection accuracy associated with traditional methods. By leveraging existing image datasets, the system eliminates the need for extensive laboratory testing while offering a cost-effective and efficient solution. This technology simplifies plant disease identification, providing smallholder farmers with a practical tool to enhance crop productivity and address agricultural challenges effectively.

2. MATERIALS AND METHODS:

This section outlines the methodology used to develop and implement the plant disease classifier. The classification process using Convolutional Neural Networks (CNN) is divided into three phases, each addressing distinct tasks to ensure an accurate and efficient system. All experiments and implementations in this study were conducted on a single machine, with the system specifications detailed in Table 1.

2.1 Data Acquisition:

The datasets for this study were obtained from publicly accessible repositories. Images of Potato and Tomato plants were sourced from the "PlantVillage Dataset," an open-access collection that includes 54,323 images in total. Rice leaf images were acquired from the "Rice Diseases Image Dataset" hosted on Kaggle. For each plant species, relevant disease classes were chosen, as outlined in Table 2. All images were captured under controlled conditions, which may introduce a certain level of bias into the model.

To mitigate this potential bias, an additional test dataset was assembled, consisting of 50 images gathered from Google. These images include more diverse plant anatomical features, in-field backgrounds, and varying stages of disease progression. This supplementary test set is designed to assess the model's performance in real-world conditions, ensuring its robustness across a wider range of image scenarios.

Hardware & Software	Characteristics
Memory	8.0GB
Processor	Intel(R) Core™ i5-9300H CPU @ 2.40GHz
Graphics	NVIDIA GeForce RTX 2060 6GB GDDR6
Operating system	Windows 11

Table 1. Machine Specifications

2.2.Data Pre-Processing:

The dataset was split into two portions, with 80% allocated for training and 20% reserved for validation. To enhance the robustness of the training process, data augmentation techniques were applied dynamically during each epoch. These augmentations were probabilistically weighted and included operations such as random flipping, padding with reflection mode, and zooming with cropping (scale = (1.0, 1.5)). However, the zoom-with-crop technique was later discarded because it inadvertently cropped crucial portions of infected leaf areas, potentially affecting the accuracy of the model.

After augmentation, all images were resized to a standard dimension of 150 x 150 pixels using a compress function. Following resizing, the images were normalized using RGB ImageNet statistics, ensuring compatibility with the pre-trained model utilized in the study.

This normalization step aligns the dataset with the model's requirements for consistent and efficient processing. A visual sample of the final preprocessed images is shown in Fig. 1.

2.3. Classification by CNN:

2.3.1 Phase One – Effect of Image Size on Model Performance

The objective of Phase One is to evaluate how different image sizes impact the performance of the model. A total of five image sizes, ranging from 150 x 150 to 255 x 255, were tested during this phase. Initially, the pre-trained ResNet34 model weights were downloaded. In the transfer learning process, all layers, except for the final two, were frozen. These unfrozen layers were modified with new weights specifically tailored for the plant disease classification task. Freezing the earlier layers ensures that the gradients do not backpropagate through them, allowing targeted training of the newly added layers.

To train these final layers, the 1cycle policy was employed. After completing this step, the previously frozen layers were unfrozen to enable fine-tuning of the entire network. A learning rate vs. loss plot was generated and analyzed during this stage to identify an optimal learning rate for fine-tuning. Once the training process was completed, the model's performance was recorded.

The experiment was then repeated for the remaining four image sizes listed in Table 2. All other parameters, including the learning rate and training procedures, were kept consistent across trials to ensure a fair comparison of model performance at different image resolutions.

Table 2. Image size Trail Information

Trail	Image Size	No.Epochs	Leaner Rate
1.	150x150	4	le-05 and le-04
2.	195x195	4	le-05 and le-04
3.	224x224	4	le-05 and le-04
4.	244x244	4	le-05 and le-04
5.	255x255	4	le-05 and le-04

Table 3. Dataset used for Classification

3.3.2. Phase Two – Model Optimization.

In this phase, the ResNet34 model is optimized using the most appropriate image size. To enhance the model's performance further, additional augmentation techniques are applied (Fig. 2). These techniques include adjustments to brightness levels (0.4–0.7) and the introduction of warping (0.5).

Species	Class	No. of Images
Potato	Early blight	1000
Potato	Late blight	1000
Potato	Healthy	152
Tomato	Bacterial Spot	2119
Tomato	Mosaic Virus	160
Rice	Brown Spot	523
Rice	Leaf Blast	779
Rice	Healthy	1000

Table 3. Dataset used for Classification

Following this, the final two layers of the model are isolated and trained using the default learning rate. After completing this step, fine-tuning is performed by conducting multiple trials. These trials explore various learning rates and numbers of epochs to achieve optimal performance.

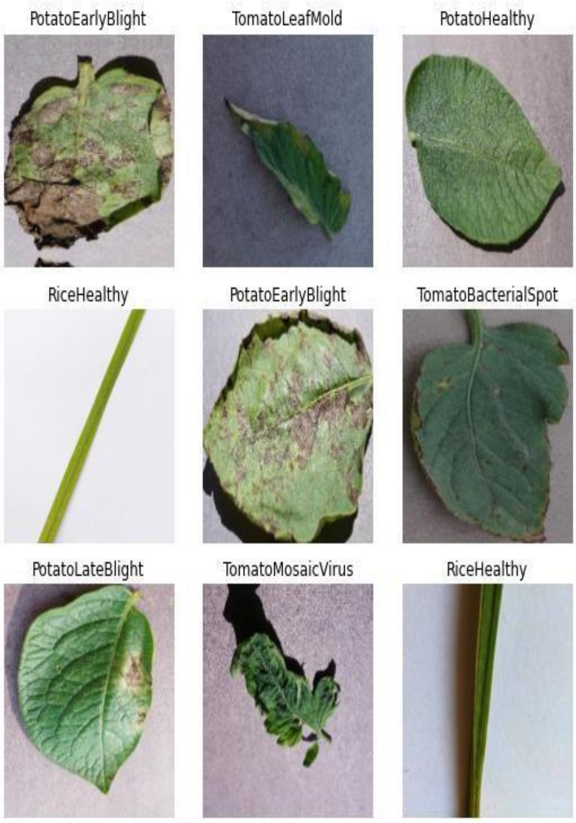

Fig. 1. Pre-processed images (Phase One): Augmentation techniques applied include random flipping and reflection padding

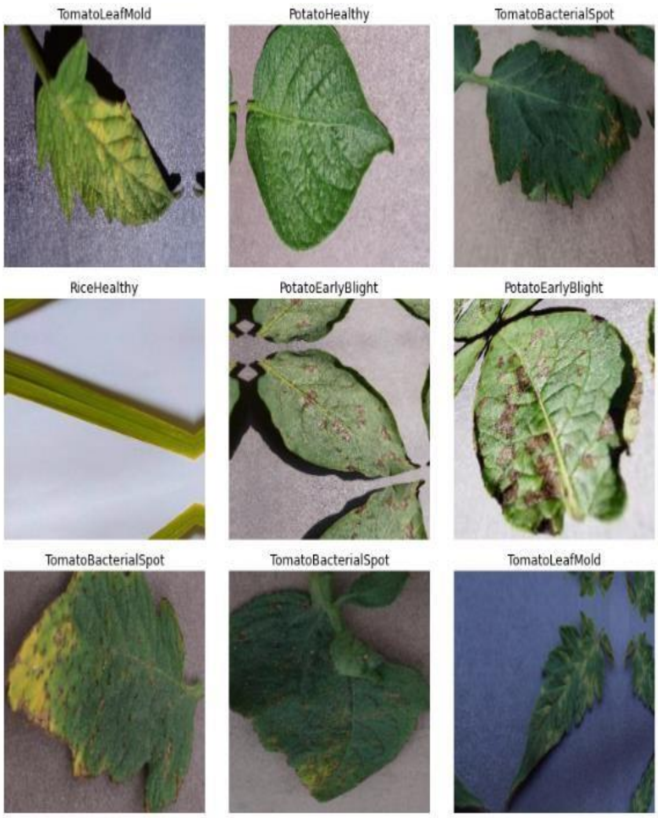

Fig. 2. Pre-processed images (Phase Two): Additional augmentation techniques include brightness adjustment (0.4–0.7), warping (0.5), random flipping, and reflection padding.

2.3.3 Phase Three – Visualizations:

To aid in interpretation, various visualizations are created using data from both the validation and test datasets. These visualizations help to analyze the model's performance and insights effectively.

The model is then deployed as part of a web application. Essential files are organized and uploaded to a GitHub repository, and the model is saved in a pickle format for deployment. The deployment process involves linking the GitHub repository to the Render platform. To facilitate this step, guidance was taken from the 'Render Examples' repository on GitHub.

3. PROJECT MANAGEMENT:

The research for this project was completed over a 12-week period, encompassing several challenging aspects that required meticulous planning and management. One notable challenge was the researcher's lack of prior experience with Python and image classification.

To establish a structured approach, both a Gantt chart and a RAID log were created online [38, 40]. These tools were initially used to define the project's scope, identify task dependencies, allocate necessary resources, and document potential risks and issues. The Gantt chart and RAID log were regularly updated and reviewed throughout the project to ensure effective progress tracking.

Due to unexpected circumstances, a two-week extension was granted, allowing all project tasks, including previously deprioritized ones, to be completed. The final versions of the Gantt chart and RAID log are included in the supplementary material file submitted separately from this report.

All programming activities were conducted using Google Colab, a free cloud platform offering a 25GB GPU. The sole expense incurred during the study was related to deploying the model on Render. The deployment, scheduled to run for one month (from April 30, 2020, to May 30, 2020), cost approximately $10. To address programming challenges, the researcher utilized online documentation for PyTorch and TensorFlow as primary references.

4. RESULTS:

4.1 Phase One – Image Size Trial

The findings from Phase One confirm that achieving an accuracy and F1 score above 90% is possible for image sizes ranging from 155 x 155 to 255 x 255. As anticipated, increasing image size enhances feature extraction but also results in longer processing times (Table 4). This preliminary analysis yielded outstanding results. As mentioned previously, the model's acceptance criterion was set at an accuracy of 80%, and these initial results significantly surpassed that threshold.

To obtain these results, models were trained with learning rates between 1e-05 and 1e-04 for four epochs. Among the tested sizes, an image size of 244 x 244 demonstrated the highest accuracy and F1 score. While existing literature (10a) commonly suggests using image sizes of 224 x 224 for plant disease classification, the results indicate that this model slightly benefits from the increased image size. Therefore, 244 x 244 was selected for subsequent phases of this research.

4.2 Phase Two – Model Optimization

Before fine-tuning, the model achieved an accuracy of 94.65% and an F1 score of 93.59% (Fig. 3). To facilitate the fine-tuning process, a plot of the learning rate (logarithmic scale) versus validation loss was analyized (Fig. 4). This analysis revealed that loss remained relatively low for learning rates between 1e-06 and 1e-04. However, beyond 1e-04, a significant increase in loss was observed.

Based on these observations, multiple trials were conducted to test the effects of learning rates. A range of 1e-05 to 1e-04 yielded the most favorable results. Fine-tuning this hyperparameter resulted in modest improvements, increasing accuracy by 1.5% and the F1 score by 1.3%. Nevertheless, the final epoch indicated slight underfitting, as evidenced by the closing training and validation metrics (Fig. 5).

To address this, the number of epochs was gradually increased. Notably, by approximately the 10th epoch, the model exhibited improved fitting, ultimately achieving an accuracy improvement of 2.8% and an F1 score increase of 3.1% (Fig. 6).

Test	Image size	Train Loss	Valid Loss	Accuracy	F1 Score	Time (hours)
1	155	0.1660	0.1222	0.9557	0.9439	2:83
2	195	0.1588	0.1150	0.9585	0.9460	3.62
3	224	0.1778	0.1256	0.9522	0.9359	4.29
4	244	0.1310	0.1153	0.9603	0.9450	5.20
5	255	0.1607	0.1249	0.9562	0.944	5.42

Table4.Results-Phase one (4Epochs,Max_lr=slice(1E-05,1E-04)

It is important to note that the validation dataset used for these evaluations comprises images with a specific layout: a single leaf against a plain background. For consistent and accurate results similar to those reported here, it is recommended to adhere to this image composition when utilizing the classifier

Fig. 3. Training the Final Layers (lr=1e-3) Prior to the fine-tuning process, the model achieved an accuracy of 94.65% and an F1 score of 93.59%.

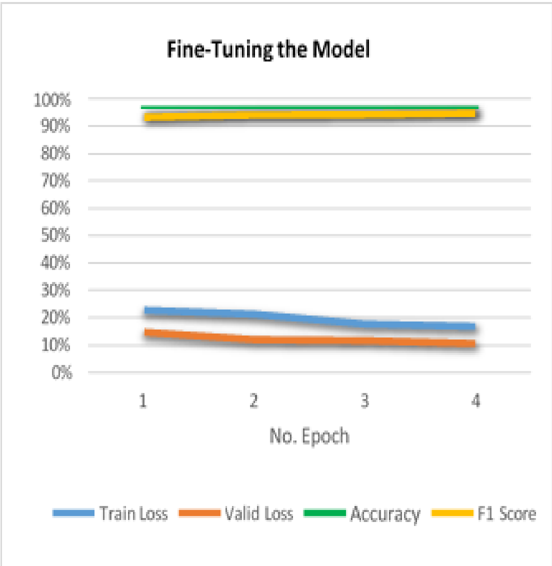

Fig. 4. Fine-Tuning the Model With a learning rate range of 1e-05 to 1e-04 and four epochs, signs of underfitting were observed during the fine-tuning process.

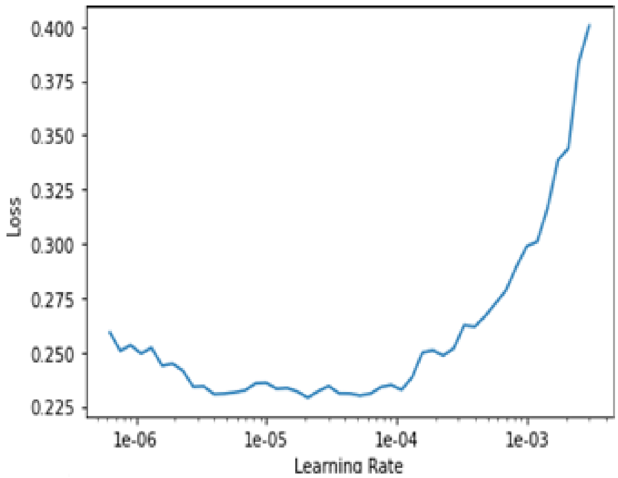

Fig. 5. Learning Rate vs. Loss The plot was utilized to guide the fine-tuning process, revealing a sharp rise in loss as the learning rate exceeded 1e-04.

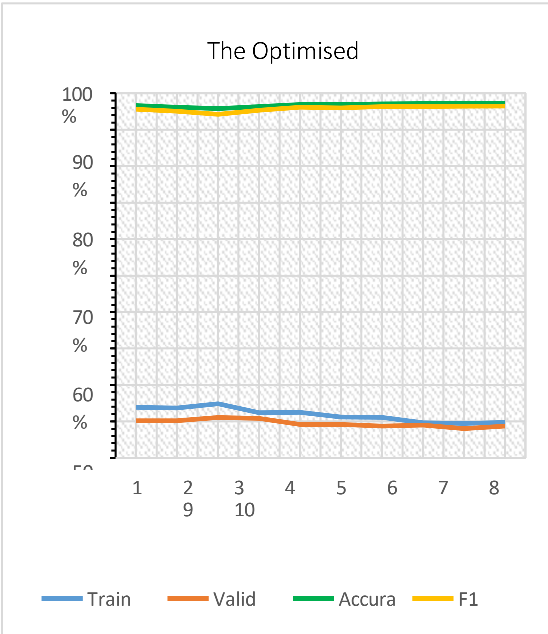

Fig. 6. The Final Optimized Model The model was optimized using a learning rate range of 1e-05 to 1e-04 over 10 epochs

4.3. Phase Three – Visualizations:

Heat map analysis offers valuable insights into the inner workings of the CNN, revealing that color, shape, and texture are key factors in identifying plant disease features. Color, in particular, plays a significant role, helping to distinguish between similar diseases by adding an additional layer of characterization. This reinforces the importance of RGB data for disease classification tasks, as discussed earlier [10, 20]. For all three species, the CNN demonstrates strong performance in recognizing disease features, with similar effectiveness observed in rice disease classes, which tend to have smaller and harder-to-identify symptoms.

The confusion matrix in Fig 6 shows the results for the validation dataset. Notably, there were no errors in the Potato or Tomato classes, while the Rice species performed poorly, indicating a potential issue with the data. Among the rice disease classes, Rice Brown Spot was the most misclassified. Approximately

13.9% of Brown Spot images were incorrectly identified as Healthy, while 9.9% were misclassified as Rice Leaf Blast.

The characteristic dark spots of Brown Spot might be confused with similar lesions in Leaf Blast, though there should be distinct differences from healthy samples. On average, 12.65% of Rice class images were misdiagnosed.

To explore this further, misclassified images were plotted and sorted by loss (Fig. 7). Upon closer inspection, it became evident that the quality of several images was subpar. Even an experienced diagnostician would struggle to make accurate assessments based on these images. These images may have been mislabeled or poorly represent the class, suggesting that they do not contribute effectively to the classifier. Consequently, such data should be excluded from the training set.

As anticipated, the model's performance declined significantly when tested on in-field images. Out of 50 images, only 44% were classified correctly. This poor performance can be attributed to several factors, including variations in plant anatomy and different background settings, which data augmentation could not address. Since the model was not trained on such data, adapting to these conditions proved challenging. Expanding the training set to include in-field imagery captured in uncontrolled environments could greatly enhance the model's robustness. As mentioned earlier, there is a notable lack of in-field plant disease imagery, underscoring the need to develop such datasets.

Fig. 7 :Using Python Flask to Create a Web Application.

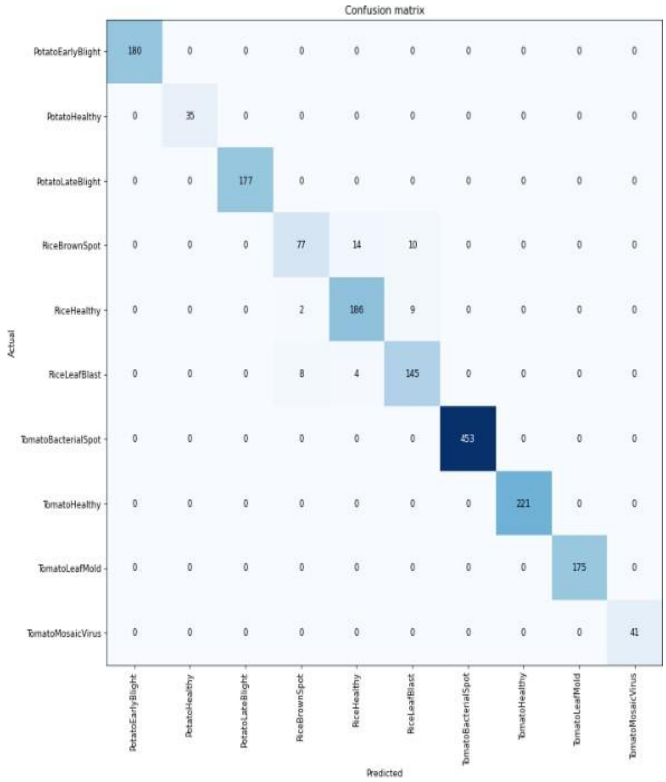

Fig. 8. Confusion matrix – validation dataset

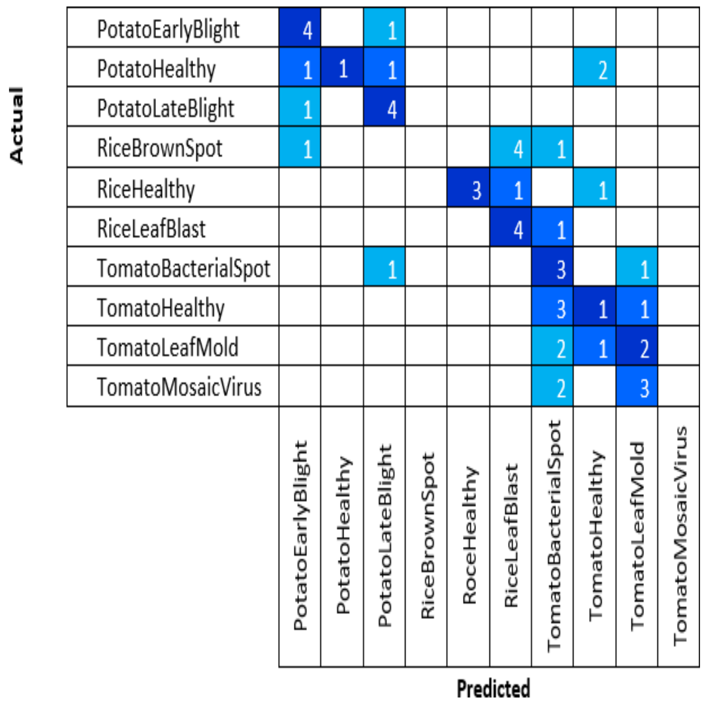

Fig. 9. Confusion matrix – test dataset

5. CONCLUSION:

Timely and accurate diagnosis of crop diseases is critical for smallholder farmers to prevent significant losses. In this study, a pre-trained Convolutional Neural Network (CNN) was fine-tuned and deployed as a plant disease detection app. This application is free, user-friendly, and requires only a smartphone and internet connection, effectively meeting the needs defined in this paper. A comprehensive analysis reveals both the strengths and limitations of the model.

When validated in a controlled environment, the model achieved an accuracy of 97.2%. This performance depends on several factors, such as the disease's stage, type, background data, and object composition. Therefore, for commercial use, a set of user guidelines is essential to ensure that the model consistently delivers this level of accuracy. Since the model was trained on images with a plain background and a single leaf, mimicking these conditions will yield the best results.

The use of augmentation and transfer learning proved valuable, allowing the CNN to generalize more effectively. Although these techniques enhanced the model's feature extraction capabilities, they were not sufficient when the model was tested on 'in-field' images. In this scenario, the classifier only achieved an accuracy of 44%. This result emphasizes the need to diversify the training dataset to include varied background data, additional plant anatomy, and different stages of disease progression.

In conclusion, this study effectively demonstrates the potential of CNNs to assist smallholder farmers in managing plant diseases. Future work should focus on expanding training datasets and testing similar web applications in real-world settings. Without such advancements, the challenge of controlling plant disease will persist.

REFERENCES:

1. M. Hunter, R. Smith, M. Schipanski, L. Atwood and D. Mortensen, "Agriculture in 2050: Recalibrating Targets for Sustainable Intensification", BioScience, vol. 67, no. 4, pp. 386-391, 2017. Available: 10.1093/biosci/bix010. Fig. 10. Confusion matrix – validation dataset Fig. 7. Top losses plotted – validation dataset Fig. 8. Confusion matrix – test dataset
2. "PlantVillage", Plantvillage.psu.edu, 2020. [Online]. Available: https://plantvillage.psu.edu/. [Accessed: 31- Jan- 2020].
3. D. Klauser, "Challenges in monitoring and managing plant diseases in developing countries", Journal of Plant Diseases and Protection, vol. 125, no. 3, pp. 235-237, 2018. Available: 10.1007/s41348-018-0145- 9.
4. A.Muimba-Kankolongo, Food crop production by smallholder farmers in Southern Africa. Elsevier, 2018, pp. 23-27.
5. J. Boulent, S. Foucher, J. Théau and P. St-Charles, "Convolutional Neural Networks for the Automatic Identification of Plant Diseases", Frontiers in Plant Science, vol. 10, 2019. Available: 10.3389/fpls.2019.00941.
6. "iNaturalist", iNaturalist, 2020. [Online]. Available: https://www.inaturalist.org/. [Accessed: 30- Jan- 2020].
7. I. Ltd., "Success-stories - PlantSnap: Training the world's largest plant recognition classifier. | Imagga Technologies Ltd.", Imagga.com, 2020. [Online]. Available: https://imagga.com/success-stories/plantsnapcase-study. [Accessed: 31- Jan- 2020].
8. "Smartphone users worldwide 2020 | Statista", Statista, 2020. [Online]. Available: https://www.statista.com/statistics/330695/number-ofsmartphone-users-worldwide/. [Accessed: 22- Apr- 2020].
9. K. Liakos, P. Busato, D. Moshou, S. Pearson and D. Bochtis, "Machine Learning in Agriculture: A Review", Sensors, vol. 18, no. 8, p. 2674, 2018. Available: 10.3390/s18082674.
10. Y.Toda and F. Okura, "How Convolutional Neural Networks Diagnose Plant Disease", Plant

Phenomics, vol. 2019, pp. 1-14, 2019. Available: 10.34133/2019/9237136.
11. Atabay, H. A. 2016b. A convolutional neural network with a new architecture applied on leaf classification. IIOAB J 7(5):226–331
12. A. Krizhevsky, I. Sutskever and G. Hinton, "ImageNet classification with deep convolutional neural networks", Communications of the ACM, vol. 60, no. 6, pp. 84-90, 2017. Available: 10.1145/3065386.