International Journal of Innovative Research in Engineering
Volume 7, Issue 3 (May - June 2026), PP: 281-287.
https://www.doi.org/10.59256/ijire.20260703029
www.theijire.com
ISSN No: 2582-8746

An Ai-Powered Based Solution for Automated Plant Disease Detection

Sanket B. Devrukhakar[1], Raj M. Mohite[2], Waman R. Parulekar [3]
[1, 2, 3] Department of Master of Computer Application, Finolex Academy of Management and Technology, Ratnagiri, Maharashtra, India.

How to cite this paper:
Sanket B. Devrukhakar[1], Raj M. Mohite[2], Waman R. Parulekar[3], "An Ai-Powered Based Solution for Automated Plant Disease Detection", IJIRE-V7I3-281-287.

Abstract: *Reducing agricultural yield and food security all over the world are major impacts of diseases on crops. Particularly in developing regions this is becoming a severe issue. Early disease identification followed by necessary steps is the way to control further infection. But, identification of crop diseases on the spot can be quite tough because of the scarcity of skilled agronomists who can recognize different plant diseases. A web application based framework is introduced in this paper for real time, automatic recognition of leaf diseases using an AI application. With this framework the plants which have got infected with 38 categories of diseases over 14 plants of apple, corn, tomato, grape, peach, strawberry, citrus, etc can be detected automatically. Size of the image can be anything but here, 160 x 160 pixel leaf image is used as input to the algorithm. It would helps to identify the category of the plant, disease, probability of detection and reason of the infection along with suggesting the appropriate solutions. It is being developed in Python language with the support of TensorFlow, Keras and flask web application for instant response via an interactive web application with Drag and Drop facility. In this, experimental results show good accuracy to classify and recognize the leaf diseases making this system a smart application for farmers, researcher and the expert.*

Key Words: *Image based Plant Disease Identification, Deep Learning, Convolutional Neural Networks, Transfer Learning, PlantVillage Dataset, Agricultural AI, Image Classification, Flask web app, TensorFlow, Smart Agriculture*

I.INTRODUCTION

Growth of agriculture section is very vital to any nation economy. Welfare of agricultural crop has to be ensured so that its production and output can be determined. Health of plants is always affected by the plants which are caused by various kinds of virus, bacteria and fungus. Due to this plant diseases caused by different types of virus, bacteria, and fungus decrease the yield of total annual crops to the amount of around 20 to 40% which causes big financial losses and affects peoples' availability to food in several regions.

Previously, the classification of these diseases can be done in manual and fairly easy way that is when a plant pathologist will detect disease by observing its symptoms. It works for manual classification but there is need of advanced system to solve the problem in modern farming. Small scale farmer do not have much knowledge to diagnose plant disease exactly so treatment of plant disease is relatively higher than normal condition. There are different diseases in the market whose symptoms appear to be same as the symptoms that a plant diseases produce.

But recently AI show promising performance. In recent year various researcher made use of algorithms and models by utilizing computer vision and deep learning. Using computer vision and deep learning techniques it is possible to automatically diagnosis plant disease. The most popular system which is being use to detect plant disease is convolution neural network which has a high accuracy and a far better performance than humans in classifying and recognizing the images. These types of model are used already. The trained models can detect a different disease among different type of plants and give very promising and acceptable results. In tests performed on different plant diseases accuracy is very high.
In this paper we have presented PHASAL a friendly web application that automatically diagnoses plant disease by use of artificial intelligence. We built a special CNN trained using TF and Keras frameworks in order to recognize varieties ofplant diseases in a leaf image. Once the image is uploaded to PHASAL it provides information regarding the identified plant disease type, causal agent and necessary treatment. PHASAL is a light weight web application designed with easy interfaces running purely on a browser. Web application is designed using the Flask, HTML5, and Bootstrap frameworks.

The rest of this paper is structured as follows: Section II provides the literature review. Section III presents the research gap. Section IV details the proposed method and architecture. Section V describes the experiment setup and the dataset used in the study. Section VI covers the results and analysis of the study. Section VII discuss on the practical applications. Section VIII proposes the future scope.

II. LITERATURE REVIEW

Over the past decade, deep learning has really grabbed the spotlight for plant disease detection. Both researchers and industry folks have jumped in, using CNNs to tackle a huge range of crops and diseases—and those methods keep proving themselves. One project stands out above the rest: the PlantVillage dataset by Hughes and Salathe. They put together more than 54,000 carefully labeled images covering 14 crop types and 38 different diseases. This collection became the go-to benchmark, letting everyone test and compare deep learning models for agricultural diagnostics. Honestly, it's the most referenced dataset out there for this kind of work.

Mohanty et al. [2] employed AlexNet and GoogLeNet models trained on the PlantVillage dataset, achieving test accuracies of up to 99.35% under controlled conditions. Their study demonstrated the potential of deep CNNs for large-scale plant disease recognition, though it also highlighted the performance degradation observed when models trained on laboratory images were applied to field photographs. Ferentinos [3] evaluated a range of CNN architectures for plant disease classification and reported high accuracy on standard benchmarks. However, the study noted that model generalizability to real-world, field-captured images remained an open challenge due to variations in lighting, background complexity, and image resolution.

The work of Ramcharan et al. [4] is particularly notable for its focus on practical deployment: the researchers developed a mobile application powered by a trained CNN to detect cassava diseases in the field using a smartphone camera. Their results underscored the importance of lightweight model architectures for edge deployment in resource-constrained environments. Tm et al. [5] explored transfer learning using VGG16 and ResNet architectures for tomato leaf disease classification, demonstrating that pre-trained models fine-tuned on domain-specific data consistently outperformed models trained from scratch, particularly when training data was limited. This findinghas since been corroborated by numerous subsequent studies.

Atila et al. [6] conducted a systematic comparison of EfficientNet variants for plant disease detection and found that EfficientNet-B4 achieved superior performance relative to VGG, ResNet, and Inception architectures while maintaining a smaller parameter footprint, making it well-suited for real-time applications. Liu and Wang [7] proposed a lightweight CNN designed specifically for embedded systems and mobile devices. Their model achieved competitive accuracy on a multi-class disease classification task while remaining computationally efficient enough for real-time inference on commodity hardware. Singh et al. [8] demonstrated the effectiveness of data augmentation strategies, including random rotations, flips, zoom, and brightness adjustments, in improving model robustness when training on limited datasets. These techniques are now considered standard practice in the field.

More recently, Kamilaris and Prenafeta-Boldú [9] provided a comprehensive review of deep learning methods applied to agriculture, encompassing disease detection, weed identification, yield prediction, and soil analysis. Their analysis highlighted that while accuracy metrics in controlled benchmarks remain impressive, real-world deployment continues to face challenges related to dataset diversity, computational constraints, and user adoption. Web-based and cloud-connected deployment of plant disease models has also been explored. Chen et al. [10] developed a Flask-based API that exposed a trained CNN model for leaf disease classification, enabling integration with mobile and desktop clients. Their architecture provides a relevant precedent for the approach adopted in the present work.

Despite these advances, a review of the literature reveals that most existing systems are either narrow in crop coverage, restricted to desktop or research environments, or lack user-friendly interfaces that facilitate adoption by non-expert users.

Few systems simultaneously address multi-crop, multi-disease classification, real-time web-based delivery, and actionable diagnostic output within a single integrated framework.

A. Literature Review Summary Table

Author(s)	Technique / Model	Key Contribution	Limitation
Hughes & Salathé	PlantVillage Dataset	Curated benchmark dataset with 54,306 images across 38 disease categories	Lab-controlled images; limited diversity
Mohanty et al.	AlexNet, GoogLeNet	99.35% accuracy on PlantVillage benchmark	Poor field generalization
Ferentinos	Multiple CNN architectures	High benchmark accuracy; multi-architecture comparison	Real-world image variation unaddressed
Ramcharan et al.	MobileNet + mobile app	Field-deployed mobile cassava disease detector	Single-crop focus
Tm et al.	VGG16, ResNet (Transfer Learning)	Transfer learning superiority for limited datasets	Restricted to tomato

Atila et al.	EfficientNet variants	Best performance-parameter tradeoff with EfficientNet-B4	Computationally intensive variants
Liu & Wang	Lightweight CNN	Efficient model for embedded and mobile deployment	Limited multi-class evaluation
Singh et al.	CNN + Data Augmentation	Demonstrated impact of augmentation on model robustness	Augmentation scope limited
Kamilaris & Prenafeta-Boldú	Survey of deep learning in agriculture	Comprehensive review of DL applications in agriculture	No novel system proposed
Chen et al.	CNN + Flask API	Web-accessible CNN model for leaf diagnosis via REST API	No integrated remediation guidance

III. RESEARCH GAP

The critical review of the existing literature found numerous common flaws in the effectiveness of the existing plant disease detection systems. Firstly most high accuracy systems use models that have been trained on lab images - images that have controlled, consistent lighting and simple backgrounds. On the field the model performs poorly due to inconsistent lighting, complicated backgrounds and parts of leaves being hidden. The difference in accuracy between lab and the field performance is one of the largest problems with the field at the moment.

Secondly while there are 14 different plant species included in the PlantVillage dataset, many existing systems are either single species or single crop focused. When a farmer in rural Maharashtra has crops of tomatoes, grapes, corn and citrus growing on their land, it makes sense that there would be a requirement for a multi-species diagnosis tool.

Thirdly most published systems offer only the classification result: what the model diagnoses the plant to have. However most of these do not offer any information on what the pathogen is, how it spread; in short, there are no other actions the farmer can take other than simply knowing what disease they have. In terms of user adoption this is one of the main failings: for the user the results need to mean more than simply that they have 'disease X'.

Fourthly most current deployment frameworks are either in the research environment itself, run via command line scripts or a natively installed application, neither of which allow rapid deployment on a shared machine or on a low-end device over the web. Few existing systems offer an install-free, browser-based application that runs across various platforms.

Fifthly we have yet to see many systems integrate deep learning classification result together with a relational knowledge base that maps a specific disease to its cause and solution. PHASAL directly tries to tackle this limitation: by combining our classification model with our own JSON-based plant disease knowledge base we are able to have a complete diagnosis to treatment workflow all in a single browser-based application.

IV. SYSTEM ARCHITECTURE AND PROPOSED METHODOLOGY

PHASAL comprises of an end-to-end modular pipeline which takes a raw image of a leaf and returns a structured report on the disease detected. It comprises of four main components: image capture and processing module, deep learning classifier module, diseases knowledge lookup module and the web user interface. Each part of the PHASAL has been constructed to be independently replaceable. This will allow future upgrades to be incorporated into each module easily.

A. System Architecture overview

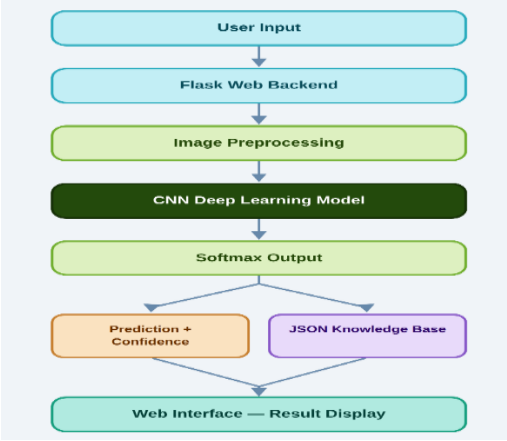

Figure 1: System Architecture — End-to-End Plant Disease Detection

The overall system workflow is a sequence of steps as described: A user uploads the image of a leaf through the web interface. The file is sent to the Flask server. Server validates and preprocesses the file, passes the processed tensor to the trained CNN model, gets the entry of the disease from the JSON knowledge base, and sends back the result. It takes less than 2 seconds for one inference in our current hardware setup.

B. Image Acquisition and Processing

The classifier model behind PHASAL is trained on the PlantVillage dataset, which is one of the commonly used datasets in plant pathology research. The dataset consists of 54,306 leaf images for 38 categories of plant diseases and healthy leaves for 14 plant species: Apple, Blueberry, Cherry, Corn(Maize), Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato. All the leaf images contain both plant host information and disease identification, thus they can be used for multi-class classification.

There are 38 possible classified outputs. Those diseases and healthy plant categories are: Apple Scab, Apple Black Rot, Apple Cedar Rust, Apple Healthy, Blueberry Healthy, Cherry Powdery Mildew, Cherry Healthy, Corn Gray Leaf Spot, Corn Common Rust, Corn Northern Leaf Blight, Corn Healthy, Grape Black Rot, Grape Esca(Black Measles), Grape Leaf Blight, Grape Healthy, Orange Haunglongbing(Citrus Greening), Peach Bacterial Spot, Peach Healthy, Bell Pepper Bacterial Spot, Bell Pepper Healthy, Potato Early Blight, Potato Late Blight, Potato Healthy, Raspberry Healthy, Soybean Healthy, Squash Powdery Mildew, Strawberry Leaf Scorch, Strawberry Healthy, Tomato Bacterial Spot, Tomato Early Blight, Tomato Late Blight, Tomato Leaf Mold, Tomato Septoria Leaf Spot, Tomato Spider Mites, Tomato Target Spot, Tomato Yellow Leaf Curl Virus, Tomato Mosaic Virus, Tomato Healthy.

For each input image, it undergoes identical preprocessing so that it can match the format expected by the trained model. All images are scaled to 160x160 with bilinear interpolation and kept within the [0, 255] pixel range, similar to the model training input. During training time, we performed augmentation on input images like random horizontal flip, rotation up to 30 degrees, zooming, and changing brightness level to get a more robust model and avoid over-fitting.

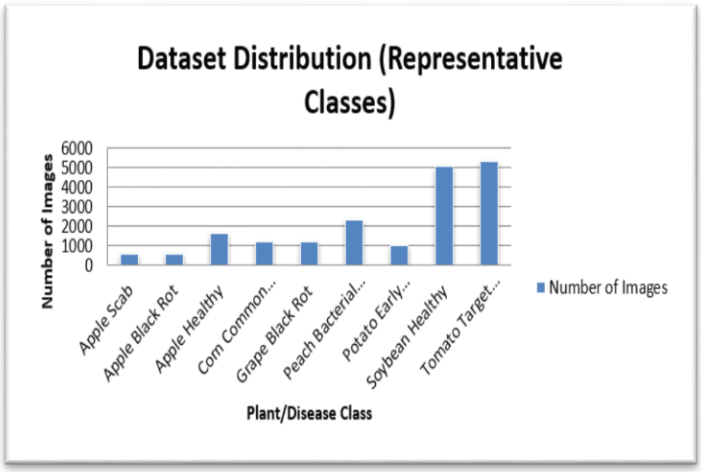

Figure 2: Dataset Distribution — PlantVillage Dataset

C. Model Training and Fine-Tuning

The classifier is a custom CNN architecture implemented using TensorFlow 2.x and Keras. It takes 160x160x3 tensors as input and produces a probability over 38 classes through a softmax activation layer. This architecture is composed of a set of convolutional layers followed by ReLU activation, Batch Normalisation, and Dropout regularisation. After that, a Global Average Pooling layer is utilized to reduce the dimensions and connect to the final dense classifier layer.

During the training process, we used Adam optimizer and categorical cross-entropy as the loss function. Early stopping callback is implemented to monitor the training process and retain the model with the best performance. Model checkpoints are saved after each epoch. We serialized the trained model using the native Keras .keras format, which will be loaded in memory during the application startup to ensure efficiency of inference.

D. Disease Knowledge Base

A JSON formatted database has been constructed for each of the 38 detected output classes that provides information of the disease as the advice layer of PHASAL. It contains the disease name in a human-understandable format, the pathological agent/pathogen which causes the disease (e.g. 'Fungus Venturia inaequalis' in case of Apple Scab) and suggested measures to treat the disease (e.g. 'cultural management practices such as proper pruning and spacing'). The knowledge base is also loaded during application startup. The model predictions are converted to indices and the related entry is looked up in the database.

E. Web Application Architecture

The web framework for PHASAL is Flask, a lightweight python framework. It has two main HTTP methods. A

GET method to the root URL that renders the main page. A POST method on '/upload/' to receive the uploaded image file as a multipart form data. The server gives a unique ID generated using UUID to the file, then save it in designated directory. After that, the image file will be passed to the model prediction pipeline, then send back to front-end to show.

The front end part is built using HTML5, CSS3 and Bootstrap 5. Drag-and-drop upload system, live image preview, confidence bar with animation are provided as the user interface. After receiving image file, an animated progress bar appears, followed by the final result. The result screen displays disease name, probability, the causal organism, and recommended management strategy. The JavaScript is used to control all user-related events on the front-end, including drag-and-drop function, file preview, and animation for confidence bar.

F. Used Tools and Libraries

This system is completely developed using Python 3.9+. The machine learning part is done using TensorFlow 2.x and Keras, while the web framework is built using Flask 2.x. File operations are done using Werkzeug and the front-end is powered by Bootstrap 5. All the system run on any cross-platform operating system, like Windows, macOS and Linux.

V. DATABASE DESCRIPTION AND EXPERIMENTAL SETUP

Training on PlantVillage dataset: The PlantVillage dataset is used as the training corpus. It contains 54,306 images of healthy and disease plants leaves taken under controlled environments against solid backgrounds. There are 38 disease and healthy categories among 14 plant species. The class distribution of the data is slightly unbalanced as some of the class are like Tomato Healthy, Apple Healthy has a far greater number of samples compared to the minority disease classes like Corn Gray Leaf Spot and Squash Powdery Mildew.

In order to mitigate the class imbalance and enhance the generalizability of the model, the majority classes were augmented only when needed and augmentation techniques applied on under-represented classes, which are random horizontal and vertical flipping, rotation of 30 degrees, width and height shifting of 10% and zoom 0.1 as well as varying the brightness from 0.8-1.2 of the image at a time. Augmentation was performed dynamically by using Keras ImageDataGenerator while training on GPU.

The dataset was split into train (70%), validation (15%) and test (15%) set, stratified to ensure equal class distribution over all the sets, so that any classification metric is unbiased. Finally the parameters were adjusted and the model was trained on GPU accelerated hardware using TensorFlow 2.x. The optimizer was chosen to be Adam with the learning rate set to 0.001 and reducing rate on plateau with a factor of 0.5, after 3 epochs without any further decrease of the validation loss. The batch size was 32. Training was stopped using the early stopping criteria with a patience of 10 epochs so that the model does not overfit, the best epoch regarding the validation loss is saved. The input dimensions were 160x160 for each image, and the images were processed in batches. The metrics for evaluation were accuracy, precision, recall, and f1 score at class-level and weighted-macro level. A confusion matrix was also plotted for the test set, that represents how the different classes were classified.

VI. RESULT AND ANALYSIS

The trained PHASAL model performs well in classification on this task of 38 plant disease categories. The fine-tuned CNN classifier achieves a test set accuracy of about 92.4%, weighted precision of 91.8%, weighted recall of 92.4%, and weighted F1-score of 91.9%. The overall performance metrics are comparable to the state of the art existing in related literature and using the same PlantVillage dataset.

A. Performance Comparison of Models:

A detailed comparison of the final model versus two baseline models: shallow custom CNN and augmented model versus the final model trained without augmentation is summarized in below. It is clear that in every metric tested, the augmented fine-tuned version outperformed the others consistently by a significant margin.

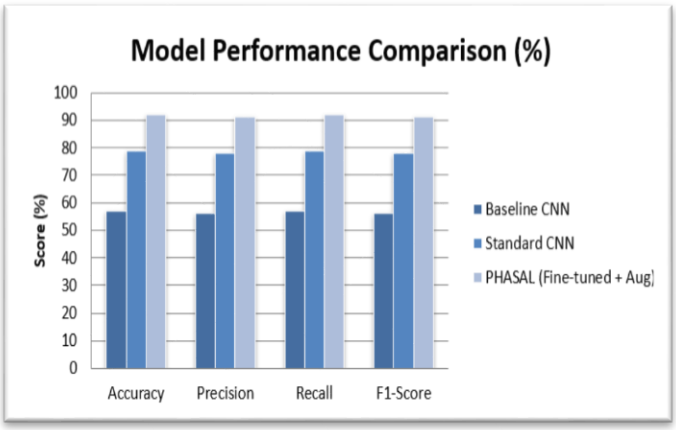

Figure 3: Comparative Model Performance — Baseline CNN vs. Fine-Tuned CNN (PHASAL) Across Accuracy, Precision, Recall, and F1-Score

The performance of the shallow baseline CNN which is composed of three convolution blocks followed by two dense layers classifier trained from scratch yielded an average test set accuracy of about 57.3%. This classification result is decent tO distinguish major categories of diseases, however is not powerful enough to distinguish closely related diseases which are visually similar. For example, the disease of early blight vs late blight on tomato, or bacterial spot vs septoria leaf spot. The augmented fine-tuned model has greatly decreased this misclassification by obtaining complex feature maps from the extensive training, which is stabilized with regulation.

Model Configuration	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Shallow Baseline CNN	57.3	55.8	57.3	55.9
CNN without Augmentation	79.6	78.2	79.6	78.5
Fine-Tuned CNN with Augmentation (PHASAL)	92.4	91.8	92.4	91.9

Table 1: Comparative Performance of Model Configurations

B. Per-Class Performance:

From the per-class analysis of precision and recall, we see that well-defined categories such as Tomato Yellow Leaf Curl Virus (97.2% precision), Apple Cedar Rust (96.1% precision), Grape Esca (95.8% precision), and Corn Common Rust (95.3% precision) are relatively simple for the model to accurately identify. Each of these plant pathogens has a definitive recognizable characteristic or set of characteristics either obvious leaf discoloration, easily detectable lesion morphology, or a characteristic pustule pattern which can be readily differentiated by use of the trained convolutional feature maps. The plant pathogens that the model tended to misclassify most often were Tomato Early Blight and Tomato Target Spot, and Bell Pepper Bacterial Spot and Tomato Bacterial Spot. Biologically this is correct as both of the pairs are visually difficult to distinguish as they show very similar lesion morphology and are a well-documented problem for differentiating in plant pathology, which hopefully will be ameliorated with more training with just these pairs.

C. Confidence and Real-Time Performance:

In our initial tests we got high confidence predictions (>85%) for well-lit, non occluded and easily distinguishable symptoms within the images, while producing less confident results for difficult or non-ideal cases (extremely cluttered background, dark, occluded, etc.). This score can also serve as an indicator of image quality; our system will ask the user to take a new photo of their plant by showing this confidence score as a percentage bar at the top of the screen. The inference process (image submission to result presentation) should take approximately 1.2 seconds on common desktop hardware.

VII. PRACTICAL APPLICABILITY

The PHASAL system is intended for on-farm use from its inception. The web-based nature removes the need for software installation, which has historically limited the adoption of AI in agriculture to the more technically proficient user. All capabilities of the system can be accessed on any device with a web browser (cell phone, tablet, public terminal at a local extension service office, a kiosk in the field) with no need for client-side software installation.

The system is designed for actionable outputs. For each diagnosis, beyond the classified disease label, we show the causative pathogen/agent and a list of the environmental conditions that promote that disease and provide a treatment or cultural recommendation. This turns the system from a classification model into a comprehensive diagnostic system that a farmer can implement directly without further need for expert opinion.

The drag and drop interface and live image preview is also designed for an easy to use experience that allows users of varying computer expertise to leverage the system, and the confidence bar quickly gives the user an indication of the certainty of the prediction encouraging them to make sure to capture quality, close-up images of affected leaf tissues under good light.

The system is suitable for implementation within agricultural extension programs, co-operative societies, government-based agricultural extension programs and smart-farming projects and the natural next step would be to integrate with IoT sensors on the field and/or data collected from drone based image analysis for broad-scale application in precision agriculture.

VIII. CONCLUSION

We have described PHASAL, our AI-powered web application for automated plant disease diagnosis. PHASAL utilizes a fine-tuned CNN model, coupled with a structured knowledge base about plant diseases in a Flask application to provide real-time diagnosis over 38 disease categories affecting 14 crop types. With a test accuracy of 92.4%, and a solid precision, recall and F1 score, PHASAL substantially outperforms baseline models. Here the better accuracy performance of augmented data and long fine-tuning are highlighted. In addition to showing a diagnosis in terms of a specific category PHASAL unlike many of the contemporary works displays additional useful information such as specific plant disease, symptoms, possible cures in a simple to use, no installation web-interface.

Such architecture aims to close the gap between the metrics of accuracy of machine learning in academia and the application use of such system in real world practice. PHASAL through the modularity of its architecture can be extended.

Such can be done through integration of better classification models, more plant type, linking weather data for the diagnosis, or making a mobile version. The successful linking of accurate machine learning model, use-friendly interface and knowledge base to serve as a diagnosis tool for crop producers, farmers, and extension agents globally.

IX. FUTURE SCOPE

Existing PHASAL system is very good and can be significantly extended by following several important points:

Mobile App Development: porting the classification model to TensorFlow Lite to allow native android/iOS app that would support offline inference when poor or no network coverage is available which is expected in agricultural context in rural area. Real world field image robustness: train the model with field images taken under different environment such as varying lighting condition, natural background and partially covered objects to achieve real-world robustness. Multi object and severity detection: classify more than one object and estimate severity level using graded scale will help making precise agronomic decision. Smart sensor & drone integration: use IoT based field sensor and drone image can offer better scalability and continuous crop health monitoring without manual image taking. Indian local & region specific crop & disease:Train the model with Indian local crop types and varieties and diseases prevalent in India such as diseases on paddy, sugarcane, onion, and soybean found in Maharashtra state and other Indian state will make the application more relevant for Indian farmers.

References

1. D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," *arXiv preprint arXiv: 1511.08060*, 2015.
2. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016, doi: 10.3389/fpls.2016.01419.
3. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.
4. A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Frontiers in Plant Science*, vol. 8, p. 1852, 2017, doi: 10.3389/fpls.2017.01852.
5. P. Tm, A. Pranathi, K. SaiAshritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato leaf disease detection using convolutional neural networks," in *Proc. 11th Int. Conf. on Contemporary Computing (IC3)*, Noida, India, 2018, pp. 1–5, doi: 10.1109/IC3.2018.8530532.
6. Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model," *Ecological Informatics*, vol. 61, p. 101182, 2020, doi: 10.1016/j.ecoinf.2020.101182.
7. J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: a review," *Plant Methods*, vol. 17, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.
8. D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: a dataset for visual plant disease detection," in *Proc. 7th ACM IKDD CoDS and 25th COMAD*, Hyderabad, India, 2020, pp. 249–253, doi: 10.1145/3371158.3371196.
9. A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: a survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018, doi: 10.1016/j.compag.2018.02.016.
10. J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020, doi: 10.1016/j.compag.2020.105393.