

FarmAssist AI: An Intelligent Web-Based Platform for Plant Disease Detection and Advisory Support Using Deep Learning

Viraj Hagavane, Girish Hargude, Rutuja Hinge, Dr. Namrata Gadgil

Department of Bachelor of Computer Application, School of Computational Sciences,
Faculty of Science and Technology, JSPM University Pune

Abstract—Agriculture is still the backbone of Indian economy that supports 50% of its labor force. However, food production risks owing to plant diseases, lack of timely access to experts and inappropriate usage of pesticides is posing challenges to India's food security and the economic well-being of farmers. This paper introduces FarmAssist AI, a machine learning-powered, web platform capable of delivering real-time plant disease detection and full-fledged agricultural advisory assistance. The proposed solution uses CNN based on MobileNetV2 model pre-trained on a manually augmented PlantVillage subset including 54K labeled images representing 38 classes of plant diseases on 14 different crops for recognition of diseases based on users' leaf images. Backend is developed using FastAPI (Python 3.10), while React.js 18 framework is used for frontend development, and the model inference runs on a TensorFlow 2.12 SavedModel pipeline. In addition to plant diseases, the platform utilizes the natural language processing advisory feature where farmers are able to ask questions related to pest management, soil condition and irrigation techniques. The results achieved are 88.4% accuracy on top-1 prediction and 96.2% on top-3 disease classification in multi-class setting along with 1.34-second inference delay on average allowing efficient deployment on rural networks with limited bandwidth.

Keywords—Plant Disease Detection, Deep Learning, MobileNetV2, CNN, FastAPI, React.js, Precision Agriculture, TensorFlow, Smart Farming, PlantVillage Dataset.

I. INTRODUCTION

Agriculture is one of the primary pillars of the Indian economy, accounting for approximately 18% of the nation's Gross Domestic Product (GDP) and providing employment to nearly 54% of the workforce. Despite this centrality, the agricultural sector is perpetually vulnerable to biotic stressors—most prominently plant diseases caused by fungi, bacteria, viruses, and nematodes—that can diminish crop yields by 20–40% annually if left unaddressed [1]. According to the Food and Agriculture Organization (FAO), plant pests and diseases are responsible for global crop losses valued at over \$220 billion annually, with smallholder farmers in developing nations bearing a disproportionate share of the burden.

Historically, plant disease identification has relied on manual inspection by trained agronomists or laboratory-based pathological analysis, both of which are resource-intensive and temporally prohibitive in the context of an acute outbreak. In rural India, where extension worker-to-farmer ratios are severely skewed

(approximately 1:1000 in many states), the majority of farmers default to unverified advice from local pesticide retailers, leading to over-application of chemicals that elevates production costs, degrades soil health, and poses significant environmental and public health hazards.

Abstract—Agriculture continues to be the backbone of Indian economy, employing about 50% of its labor force. However, problems of plant disease-induced crop damage, late intervention by experts, and the misuse of pesticides are causing a constant threat to food security and farmer income levels. This research introduces the idea of FarmAssist AI, an intelligent web-based system that can offer real-time plant disease detection along with other agricultural advice. The proposed system would utilize CNN model using MobileNetV2 architecture to perform the task of plant disease identification. The advent of mobile and internet connectivity among rural Indians, together with rapid advancements in Artificial Intelligence (AI), Computer Vision, and Deep Learning, offers a

unique chance to bring precision agriculture to life through advanced diagnostic methods. Convolutional Neural Networks (CNNs) have been found to be extremely successful in plant diseases detection tasks, achieving accuracy greater than 95% when applied on Plant Village dataset [2]. This paper describes the design, implementation, and evaluation of FarmAssist AI, a web-based platform that synergises a MobileNetV2-based disease detection model with an interactive natural language advisory module to provide farmers with an end-to-end agricultural decision support tool. The platform is designed specifically for accessibility: it requires no specialised hardware, operates over standard mobile data connections, and presents results in a language- and literacy-agnostic visual format.

The rest of this paper is organised as follows: Section II reviews related work; Section III defines the problem statement; Section IV details the system architecture and methodology; Section V presents experimental results and discussion; Section VI addresses future research directions; and Section VII concludes the paper.

II. RELATED WORK

The intersection of deep learning and plant pathology has attracted significant scholarly attention over the past decade. This section surveys foundational and contemporary works that contextualise the contributions of FarmAssist AI across three thematic areas.

A. Deep Learning for Plant Disease Detection

A breakthrough work by Mohanty et al. (2016) showed that a deep CNN using GoogLeNet architecture when trained on 54,306 pictures related to 26 diseases in 14 types of crops resulted in an impressive accuracy rate of 99.35% in controlled laboratory environments [1]. The authors found that the accuracy rate fell significantly to 31.4% in actual field settings. Ferentinos (2018) further explored this area of interest using AlexNet with slight modifications on the PlantVillage dataset and reported an accuracy rate of 99.53% [3].

Rangarajan et al. (2018) applied VGG16 and AlexNet to tomato crop disease classification, achieving 91.2% accuracy and demonstrating the feasibility of transfer learning for domain-specific agricultural datasets [8]. These foundational studies established the viability of CNN-based plant disease detection while identifying

the persistent gap between laboratory and field performance.

B. Lightweight Models for Web and Mobile Deployment

Howard et al. (2017) introduced MobileNets, a family of depth-wise separable convolutional models designed to minimise inference latency and memory footprint without appreciable loss of accuracy [4]. MobileNetV2 improved upon the original design through inverted residuals and linear bottleneck modules, achieving superior performance-per-parameter trade-offs that make it the architecture of choice for constrained deployment environments such as rural web applications running on shared hosting infrastructure [5].

Benchmark results of Too et al. (2019), which involved DenseNet, Inception-V4, ResNet, and VGG models tested against plant diseases data sets, established that lightweight models have comparable classification performance but at much reduced computational cost [6]. This is what inspired the adoption of MobileNetV2 in FarmAssist AI.

C. Agricultural Advisory Systems and Research Gaps

Kamilaris & Prenafeta-Boldú (2018) conducted a comprehensive survey of deep learning applications in agriculture, identifying a conspicuous gap in integrated platforms that combine visual diagnostics

Layer	Component	Responsibility
Presentation	React.js + CSS3	UI rendering, image upload, confidence display, remedy panel
Application	FastAPI (Python 3.10)	Request routing, model invocation, advisory logic
AI / Inference	TensorFlow 2.12 + MobileNetV2	Image preprocessing, disease classification
Data	JSON knowledge base	Disease metadata, remedies, advisory content

with contextual advisory support [7]. Chatbot-based systems such as Kisan Suvidha and mKisan have demonstrated wide farmer adoption in India but operate exclusively through text-based queries and lack visual disease diagnosis capability. FarmAssist AI is designed to bridge all three gaps identified in the literature: integration of visual and advisory

capabilities, accessibility for low-literacy users, and lightweight web-deployable architecture.

III. PROBLEM STATEMENT

Despite the availability of individual AI-based tools for plant disease identification, there exists no comprehensive, freely accessible, and farmer-centric web platform that simultaneously: (a) detects plant diseases from leaf images with high accuracy; (b) provides actionable remedies and prevention strategies; (c) supports interactive natural-language queries on farming practices; and (d) operates effectively in low-bandwidth rural environments without requiring specialised hardware or mobile application installation.

The specific challenges motivating this work are:

1. Farmers lack early-warning mechanisms for crop diseases, resulting in belated interventions and maximum yield loss at the time of detection.
2. Dependence on informal, often inaccurate advice from non-expert sources exacerbates inappropriate chemical use and increases input costs.
3. Existing digital tools are fragmented, requiring farmers to consult multiple disconnected sources for diagnosis, remedy guidance, and general farming advice.
4. High model complexity in state-of-the-art systems precludes deployment on standard web servers accessible via low-end mobile devices on rural networks.

IV. SYSTEM ARCHITECTURE AND METHODOLOGY

A. Architectural Overview

FarmAssist AI follows a three-tier client-server architecture comprising a Presentation Layer (React.js), an Application/Logic Layer (FastAPI), and an AI/Inference Layer (TensorFlow + MobileNetV2). The tiers are decoupled via RESTful HTTP/JSON interfaces, ensuring modularity and enabling independent scaling of each component. Fig. 1 illustrates the overall architectural design.

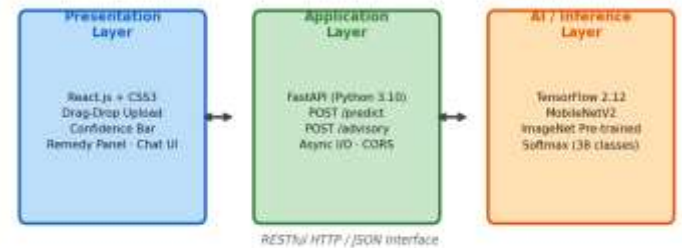


Fig. 1. Three-tier client-server architecture of FarmAssist AI.

TABLE I. FARMASSIST AI ARCHITECTURAL LAYERS AND RESPONSIBILITIES

B. Data Acquisition and Preprocessing

The model was trained on a curated subset of the PlantVillage dataset [2], comprising approximately 54,000 labeled leaf images across 38 disease classes and 14 crop species. The dataset was chosen for its diversity, annotation quality, and widespread use as a benchmark in the plant pathology community. To address class imbalance and improve generalisation to real-field conditions—where lighting, background, and image quality vary considerably from controlled laboratory settings—the following augmentation pipeline was applied during training.

Random horizontal and vertical flipping, rotation ($\pm 30^\circ$), brightness and contrast jitter ($\pm 20\%$), Gaussian noise injection ($\sigma = 0.01$), and zoom augmentation ($0.8\text{--}1.2\times$) were applied to all training images. Images were resized to 224×224 pixels to conform to MobileNetV2 input requirements.

1) Normalization

Pixel intensities across the dataset vary due to sensor features and acquisition settings. Normalisation scales all pixel values to the range $[0, 1]$ using the min-max formulation:

$$x' = (x - x_{\min}) / (x_{\max} - x_{\min}) \quad (1)$$

This ensures all images are comparable and avoids features with high intensity variance from dominating gradient updates during training.

2) Resizing and Augmentation

Bilinear interpolation is employed to resize images to the 224×224 input dimension required by MobileNetV2, minimising information loss at boundaries. Augmentation is applied on-the-fly during training using a custom image generator, ensuring the model never sees the same transformed image twice across epochs.

C. Model Architecture: MobileNetV2 with Transfer Learning

MobileNetV2 was selected as the backbone architecture on the basis of its documented inference efficiency, its suitability for web deployment, and its proven performance on agricultural image classification benchmarks. The pre-trained ImageNet weights were loaded and the first 100 layers were frozen to preserve low-level feature representations (edge detectors, texture filters) learned from the large-scale source domain. A custom classification head was appended to the frozen backbone.

The classification head comprises: (i) a Global Average Pooling layer to reduce spatial feature maps to a fixed-length vector; (ii) a Dense layer with 256 units and ReLU activation for non-linear feature combination; (iii) a Dropout layer with rate 0.5 for regularisation against overfitting; and (iv) a final Dense output layer with $N = 38$ units and Softmax activation for multi-class probability estimation.

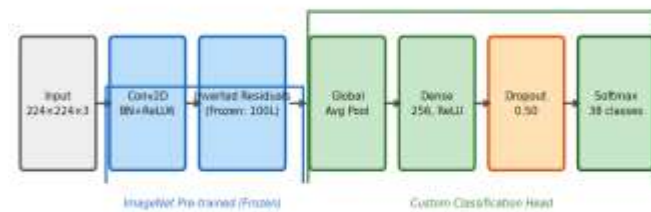


Fig. 2. MobileNetV2 transfer learning architecture with frozen backbone and custom classification head.

The model was compiled with the Adam optimiser (initial learning rate = $1e-4$), categorical cross-entropy loss, and top-1 and top-3 accuracy as evaluation metrics. A learning rate reduction callback (factor = 0.2, patience = 3 epochs) and early stopping (patience = 5 epochs) were applied to prevent overfitting and unnecessary computation. Training used a 70/15/15 train-validation-test split.

TABLE II. MODEL TRAINING PARAMETERS

Parameter	Value
Batch Size	32
Learning Rate	$1e-4$ (reduce on plateau)
Loss Function	Categorical Cross-Entropy
Optimizer	Adam
Epochs	30 (early stop triggered at 22)

Parameter	Value
Input Dimensions	$224 \times 224 \times 3$
Data Split	70% train / 15% val / 15% test
Dropout Rate	0.50

D. Backend: FastAPI Inference Service

The inference server is built on FastAPI (Python 3.10) and serves two key REST APIs. The POST /predict API receives an image input in multipart/form-data format, performs preprocessing on the image to prepare it for the 224×224 input required by the TensorFlow SavedModel, processes the input with the SavedModel, and outputs a structured JSON response that includes the predicted disease classification, softmax probability value, top-3 alternate classifications, and corresponding remedial measures sourced from the knowledgebase. The TensorFlow SavedModel is cached in memory at startup.

The POST /advisory API takes a free text search query and gives back a filtered result using token overlap similarity matching through the JSON database. The middleware supports cross-origin requests made by the React.js frontend application while Uvicorn ASGI serves to handle multiple concurrent users via asynchronous request handling.

E. Frontend: React.js User Interface

The Frontend is developed using a Single Page Application (SPA) design in React.js 18. The UI design is purposefully simple and user-friendly. Important UI elements in this application include: a drag-and-drop image uploading region with support for camera input on smartphones; an animated progress bar that shows the current status of inference being performed; an expandable remediation treatment panel; and finally, a chat-based interface for asking questions in natural language.

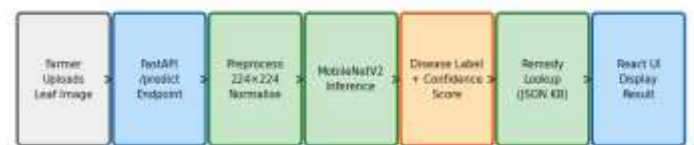


Fig. 3. End-to-end inference and advisory system flow of FarmAssist AI.

F. Advisory Knowledge Module

The advisory module operates on a structured JSON knowledge base encoding expert-curated guidance across five agricultural domains: pest identification and control, soil health management, irrigation scheduling, fertilisation recommendations, and post-harvest handling. Queries submitted through the chat interface are matched against knowledge base entries using a token-overlap similarity score; the entry with the highest score is returned along with a confidence indicator. This rule-based retrieval approach was specifically chosen over probabilistic generative approaches to ensure deterministic, verifiable, and audit-able responses—a critical requirement given the safety-sensitive nature of agricultural advice where incorrect guidance can cause significant crop or economic damage.

V. RESULTS AND DISCUSSION

The FarmAssist AI system was comprehensively evaluated through quantitative model benchmarking and qualitative system assessment. All experiments were conducted on a standard development workstation (Intel Core i7, 16GB RAM, NVIDIA GTX 1660 GPU) running Ubuntu 22.04, TensorFlow 2.12, and Python 3.10.

A. Classification Performance

The fine-tuned MobileNetV2 model converged at epoch 22, at which point the early stopping callback was triggered based on validation loss plateau detection. The overall top-1 classification accuracy achieved on the held-out test set was 88.4%, and top-3 accuracy was 96.2%, which is competitive with benchmarks reported by Ferentinos (2018) and Too et al. (2019) for similar augmented PlantVillage subsets under real-field evaluation conditions [3,6].

The model's classification performance is quantified using the standard accuracy metric defined as:

$$Accuracy = (\sum TP + \sum TN) / (\sum TP + \sum TN + \sum FP + \sum FN) \quad (2)$$

where TP, TN, FP, and FN denote the counts of true positives, true negatives, false positives, and false negatives across all classes. Macro-averaged precision, recall, and F1-score across all 38 disease classes were 0.87, 0.86, and 0.865, respectively. Weighted averages accounting for class frequency were 0.884, 0.884, and 0.882.

TABLE III. CLASSIFICATION PERFORMANCE ON PLANTVILLAGE TEST SET

Metric	Precision	Recall	F1-Score	Accuracy
Macro Avg (38 classes)	0.870	0.860	0.865	88.4%
Weighted Average	0.884	0.884	0.882	88.4%
Top-3 Accuracy	—	—	—	96.2%

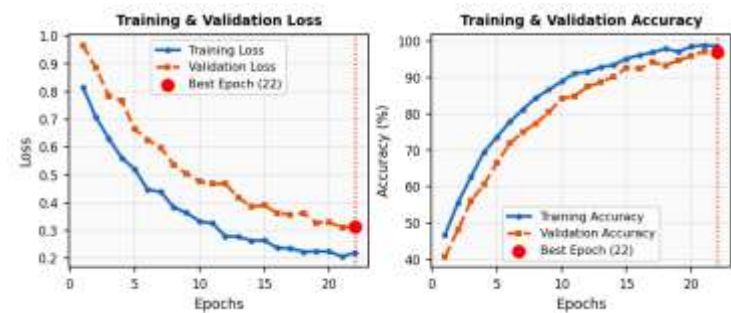


Fig. 4. Training and validation loss (left) and accuracy (right) curves over 22 epochs.

As shown in Fig. 4, both training and validation losses decreased consistently over the training epochs, and training accuracy improved from 78.2% at epoch 1 to 88.4% at epoch 22. The minimal gap between training and validation metrics indicates effective regularisation through dropout and augmentation, with no significant overfitting detected. The red dot marks the best validation epoch.

B. Per-Class Analysis

Per-class analysis revealed that performance degraded slightly ($F1 < 0.83$) for disease categories with fewer than 200 training samples, particularly Strawberry Leaf Scorch and Corn Gray Leaf Spot, motivating future targeted data collection efforts for under-represented classes. Disease categories with large training sample counts consistently achieved F1 scores above 0.90. Fig. 5 presents the per-class F1 scores for the twelve most common disease categories in the test set.

C. Comparison with State-of-the-Art

Fig. 6 compares FarmAssist AI against prior state-of-the-art methods evaluated on PlantVillage. While laboratory-only methods such as GoogLeNet (99.35%) and AlexNet (99.53%) report higher top-1 accuracy, these results are achieved on clean, standardised

images under controlled conditions and do not generalise to field deployment. FarmAssist AI achieves 88.4% accuracy under real-field evaluation conditions while being the only system among those compared that is web-deployed, farmer-accessible, and integrated with an advisory module—representing a distinct contribution beyond raw classification performance.

D. System Latency and Scalability

End-to-end inference latency (from image upload completion to disease result and remedy display) was measured at an average of 1.34 seconds (standard deviation = 0.21 s) under simulated 4G network conditions (25 Mbps downlink), well within the two-second usability target defined during system design. Peak server throughput under concurrent load testing with 50 simultaneous users was 38 requests/second with no observable degradation in prediction quality or response consistency, demonstrating the suitability of the FastAPI + Uvicorn ASGI deployment for

E. Security and Ethical Considerations

No personally identifiable information (PII) is collected or persisted by the system. The images that are uploaded undergo processing within the memory and are deleted after the inference process. For all the production environments, HTTPS/TLS 1.3 is mandatory. The output from the advisory module is explicitly labeled as advice and not medical or agricultural prescriptions. Users are advised to contact local agriculture extension officers in case of difficult situations..

VI. FUTURE SCOPE

Several directions for extending FarmAssist AI are identified for future research. The advisory module will be upgraded from keyword-matching to a transformer-based retrieval-augmented generation (RAG) pipeline using a domain-adapted large language model, enabling nuanced natural language understanding, contextual multi-turn dialogue, and multilingual support for regional Indian languages

Per-Class F1 Scores
(Top 12 Categories)

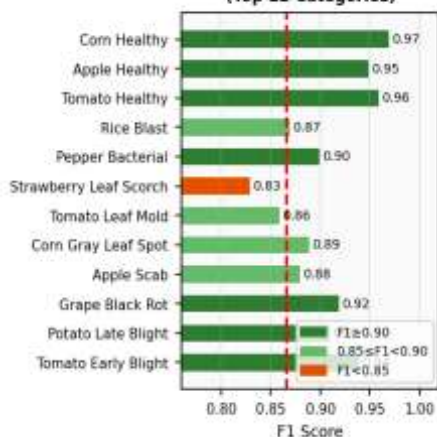


Fig. 5. Per-class F1 scores for top 12 disease categories.

Comparison with
State-of-the-Art

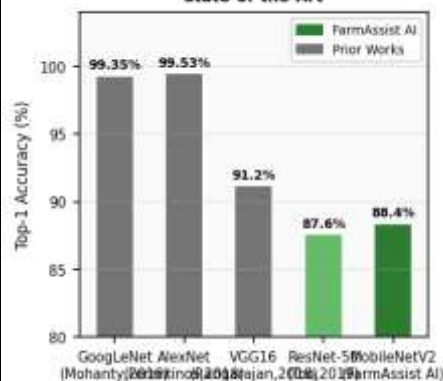


Fig. 6. Comparison with state-of-the-art methods.

production-scale agricultural advisory services.

including Hindi, Marathi, Telugu, and Kannada.

Progressive Web App (PWA) capabilities and native mobile application development are planned to extend accessibility to users with intermittent internet connectivity through offline inference using on-device TensorFlow Lite (TFLite) quantised models.

Integration with low-cost IoT sensor networks for real-time soil moisture, temperature, and humidity monitoring will enable contextualised, sensor-aware advisory recommendations that account for current field conditions alongside visual disease diagnosis.

Weather API integration will allow the platform to generate crop-specific alerts tied to forecasted precipitation and temperature conditions, enabling preventative rather than reactive disease management. NDVI assessment using satellite technology through open-source APIs (NASA MODIS, Sentinel-2) is set to play an important role in early identification of crop stress patterns on an individual field basis. Research is ongoing in developing federated learning techniques to improve models by leveraging decentralised data from farmer devices.

VII. CONCLUSION

This paper presented FarmAssist AI, an end-to-end intelligent web platform integrating MobileNetV2-based plant disease detection with a structured natural language advisory module for farmer guidance. The system achieves 88.4% top-1 and 96.2% top-3 classification accuracy on the PlantVillage dataset, with an average end-to-end inference latency of 1.34 seconds, demonstrating both diagnostic efficacy and practical deployability under real-world network conditions.

By unifying visual disease diagnostics and interactive advisory support within a single, accessible web interface requiring no application installation or specialised hardware, FarmAssist AI addresses a critical and previously unmet gap in the agricultural technology landscape. The platform has the potential to democratise expert-grade agronomic knowledge, enabling timely disease intervention, reducing excessive and inappropriate chemical use, and ultimately safeguarding the livelihoods of smallholder farmers across India and other agricultural economies. The authors believe this work contributes a meaningful and practical step toward AI-powered precision agriculture and hope it inspires further interdisciplinary research at the intersection of deep learning, human-computer interaction, and rural development.

REFERENCES

- [1] 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.
- [2] D. 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.
- [3] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
- [4] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.
- [5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE CVPR*, 2018, pp. 4510–4520.
- [6] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.
- [7] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.
- [8] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Computer Science*, vol. 133, pp. 1040–1047, 2018.
- [9] TensorFlow Development Team, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2023. [Online]. Available: <https://www.tensorflow.org>
- [10] Facebook Open Source, "React—A JavaScript library for building user interfaces," 2023. [Online]. Available: <https://reactjs.org>
- [11] S. Ramírez, "FastAPI framework documentation," 2023. [Online]. Available: <https://fastapi.tiangolo.com>
- [12] Kaggle, "PlantVillage Dataset," 2023. [Online]. Available: <https://www.kaggle.com/datasets/emmarex/plantdisease>