International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

Agriscan Ai: A Web Application for Early Detection Of Plant Leaf Diseases To Support Farmers

G. V. Rishitha[1], K. Sai Rohith[2], G. Gurumurthy[3], E. Raju[4] ,Mrs. Jahanara Begum[5]

Department Of Computer Science And Engineering Hyderabad Institute Of Technology And Management, Hyderabad, India

Abstract—Crop diseases significantly reduce agricultural productivity, and early detection is essential for preventing further damage. In practice, however, farmers largely depend on visual inspection of plants, a method that is time-consuming, error-prone and often unreliable without expert help, which is frequently unavailable in rural areas. This paper presents AgriScan AI, a web-based plant disease detection application that allows a farmer to upload or capture a photo of an affected plant leaf and receive an analysis within seconds. The proposed system uses YOLO26n, the nano model of the Ultralytics YOLO26 family, a lightweight Convolutional Neural Network (CNN)-based model, in its image-classification form (yolo26n-cls) fine-tuned on the PlantVillage dataset. The model analyses the leaf image and identifies whether the plant is healthy or diseased, and the application displays the possible disease name based on the model's prediction together with a confidence value, a severity indication and guidance on cause, treatment and prevention. The application follows the workflow CAPTURE → UPLOAD → PREPROCESS → CLASSIFY →DISPLAY → ADVISE → RECORD. Beyond classification, the prototype provides farm-management features including field mapping, scan history, a weather summary, an assistant named Krishi Mitra, an outbreak map, a farm diary and expert-help access, all within a simple and user-friendly interface. The goal is to reduce the need for manual inspection and frequent expert visits and to make disease detection faster and more accessible in remote areas. The current prototype operates in demonstration mode with illustrative results; quantitative metrics reported in this paper are illustrative prototype values and should be replaced with measured results from the trained YOLO26n model before final submission.

Index Terms—Plant disease detection, YOLO26n, convolutional neural network, deep learning, PlantVillage, precision agriculture, leaf image classification, transfer learning, web application, smart farming.

1. INTRODUCTION

Agriculture is the primary source of livelihood for a large proportion of the population in India and in many developing countries. Crop yield and quality, however, are continuously threatened by pathogens and pests. Savary et al. estimated that pathogens and pests cause substantial losses in major food crops worldwide, which directly affects food security and farm income [1]. Many of these losses can be reduced when a disease is recognised at an early stage, because timely treatment limits the spread of infection to neighbouring plants and fields.In most farms, disease identification still relies on the farmer's own visual inspection of leaves, stems and fruits.This approach requires experience, is slow when large areas must be examined, and is prone to error because several diseases, pest attacks and nutrient deficiencies produce similar-looking symptoms such as spots, discolouration and curling [2]. Agricultural experts and extension officers can provide a reliable diagnosis, but they are not always accessible, particularly in remote and rural regions where visits are infrequent and delays can allow a disease to spread.Recent progress in deep learning, and in particular Convolutional Neural Networks

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

(CNNs), has made automatic image-based diagnosis practical. CNNs learn discriminative visual features such as lesion texture, colour patterns and shape directly from images, without manual feature engineering [3]. Mohanty et al. reported high classification accuracy on a large public leaf image collection using deep networks [4], and subsequent studies confirmed that such models can be deployed for crop disease diagnosis [5]. The YOLO family of models, introduced for real-time object detection, has evolved into a set of efficient vision models that also support image classification, and its latest YOLO26 generation includes a compact nano variant suited to low-cost deployment. At the same time, widespread smartphone use among farmers makes it possible to deliver such a model through a simple web interface.

AgriScan AI is proposed as a web-based application that brings YOLO26n-based leaf analysis to farmers. A user uploads or captures a photograph of an affected leaf; the system analyses it, states whether the plant appears healthy or diseased, and displays the possible disease name along with advice. The application is intended to complement, not replace, expert advice: uncertain or severe cases are explicitly directed to a local agricultural expert.

A. Objectives

- Allow farmers to upload or capture a photo of an affected plant leaf through a web interface.
- Use the YOLO26n model, a lightweight Convolutional Neural Network (CNN)-based model trained on the PlantVillage dataset, to analyse the leaf image.
- Identify whether the plant is healthy or diseased.
- Display the possible disease name based on the model's prediction, with a confidence value.
- Reduce the need for manual inspection and frequent expert visits.
- Enable faster and more accessible disease detection, even in remote areas.
- Provide a simple, user-friendly interface suitable for farmers with limited technical experience.

2. Related Work And Problem Identification

Early computer-vision approaches to plant disease recognition relied on hand-crafted features such as colour histograms, texture descriptors and shape measures combined with classifiers like Support Vector Machines. Barbedo reviewed the main challenges of such visible-range image methods, including complex backgrounds, variable illumination, symptom variability and the difficulty of separating diseases with similar appearance [2]. Hand-crafted pipelines tend to perform well only under controlled conditions.Deep learning replaced manual feature design. Mohanty et al. trained AlexNet and GoogLeNet style networks on the PlantVillage collection of 54,306 images covering 14 crop species and 26 diseases (38 classes in total) and reported accuracy above 99% on held-out data, while noting that performance dropped markedly on images captured under field conditions [4], [6]. Ferentinos compared several architectures on a database of roughly 87,000 images and reported a best success rate of 99.53% [5]. Too et al. showed that fine-tuning pre-trained networks such as DenseNet and ResNet gives strong results with relatively short training time [7], and Barbedo examined how dataset size and variety affect transfer learning for plant disease classification [8].Lightweight architectures such as MobileNet were designed for resource-constrained devices and make on-device or low-cost server inference possible [9], [10]. Transfer learning, in which a network pre-trained on a large dataset is adapted to a new task, further reduces data and training requirements [11].The YOLO (You Only Look Once) family was introduced by Redmon et al. as a

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

single-pass, real-time object detector[12] and has since been extended by the Ultralytics framework to detection, segmentation, pose estimation and image classification. The YOLO26 generation offers models in several sizes; its classification variants carry the suffix -cls, are pre-trained on ImageNet and can be exported to deployment formats such as ONNX [13], [14]. The nano classification model, yolo26n-cls, is the smallest of them: at an input size of 224 pixels it has about 2.8 million parameters and 0.5 billion FLOPs [13], which makes it suitable for low-cost servers and for later on-device deployment. These properties motivated the choice of YOLO26n for AgriScan AI.

1. Problem Statement

Crop diseases significantly reduce agricultural productivity and early detection is key to preventing further damage. However, farmers largely rely on visual inspection to identify diseases, a method that is time-consuming, prone to error and often unreliable without expert help, which is not always available, especially in rural areas.

2. Gap Identification

Existing research demonstrates that CNNs can classify leaf diseases accurately, but many solutions are presented as isolated models or laboratory experiments that a farmer cannot use directly. Farmers also need context beyond a label: how certain the prediction is, how serious the problem is, what to do next and whom to ask if the result is uncertain. AgriScan AI addresses this gap by packaging lightweight YOLO26n-based classification into an accessible web application that returns the disease name, confidence, severity and practical guidance, and stores results so that the health of each field can be tracked over time.

3. Proposed System Architecture

AgriScan AI follows a modular web-application architecture composed of four layers. The presentation layer is a responsive web interface that lets the user upload or capture images, choose the crop and field, and read results. The application layer validates requests, handles authentication and coordinates image handling. The inference layer hosts the trained YOLO26n classification model, which receives a pre-processed leaf image and returns class probabilities. The data layer stores users, fields, scan records and diary entries.The high-level data flow is: User Interface → Application/API Layer → Image Pre-processing → YOLO26n Inference → Result Interpretation → Database and Display. The interpretation step converts the predicted class into a human-readable disease name, derives a confidence value from the softmax output, assigns a severity label and retrieves the corresponding cause, treatment and prevention text.

A. Functional Requirements

Table I lists the functional requirements of the AgriScan AI application..

Requirement	Function
FR1	Upload a leaf photo from the device or capture it with the camera
FR2	Optional selection of crop type, or automatic crop detection
FR3	Assign a scan to a specific field
FR4	Image pre-processing and YOLO26n-based analysis
FR5	Healthy versus diseased decision
FR6	Display of possible disease name with confidence value
FR7	Severity indication and possible alternative conditions
FR8	Cause, treatment and prevention guidance
FR9	Scan history and field management
FR10	Weather summary and outbreak awareness for the user's location
FR11	Conversational assistance (Krishi Mitra), listen and share options
FR12	Expert-help access for uncertain or severe cases

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

A. Yolo26n Model Design

AgriScan AI uses YOLO26n in its image-classification form, yolo26n-cls, from the Ultralytics YOLO26 family [13]. Although the YOLO name is best known from object

detection, the classification variants apply the same efficient convolutional design to whole-image labelling, which matches the PlantVillage dataset because every image carries a single class label for the leaf shown. The network is a CNN: stacked convolutional layers extract features from the RGB leaf image, from simple edges and colour patches in early layers to disease-specific lesion patterns in deeper layers, and a classification head converts the pooled features into a probability for every class. The official yolo26n-cls weights are pre-trained on ImageNet with 1000 classes [13], [14]; through transfer learning [11] the output layer is adapted to the 38 PlantVillage classes and the network is fine-tuned on leaf images. Table II summarises the model. The ImageNet figures are those published by the framework developers [13] and describe the pre-trained starting point; they are not results on plant images.

TABLE II

SPECIFICATION OF THE YOLO26N-CLS MODEL USED IN AGRISCAN AI.

Property	Value
Model variant	YOLO26n-cls (nano, classification)
Framework	Ultralytics (PyTorch)
Pre-trained weights	yolo26n-cls.pt (ImageNet, 1000 classes)
Input size	224 × 224 pixels, RGB
Parameters (ImageNet version)	about 2.8 million
Compute at 224 px	about 0.5 billion FLOPs
ImageNet top-1 / top-5 accuracy	71.4% / 90.1%
CPU inference speed (ONNX)	about 5.0 ms per image on the benchmark hardware of [13]
Output after fine-tuning	38 PlantVillage classes (healthy and diseased)
Export formats	PyTorch, ONNX, TorchScript, OpenVINO, NCNN, TensorFlow SavedModel and others

II . METHODOLOGY

The methodology is organised as a seven-stage workflow: CAPTURE → UPLOAD → PREPROCESS → CLASSIFY→ DISPLAY → ADVISE → RECORD. The farmer captures or selects a leaf photograph and uploads it; the system pre-processes and classifies the image; the result is displayed with a confidence value; the user receives treatment and prevention advice; and the scan is recorded against a field for later review.

A. Dataset

The model is trained and evaluated on the PlantVillage dataset [6], an open-access collection of 54,306 images of healthy and diseased leaves covering 14 crop species and 26 diseases, organised into 38 classes (crop and condition pairs such as Tomato Early blight, Potato Late blight, Grape Black rot and Apple healthy). Each image shows a single leaf and is annotated with its crop and condition, which suits image-

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

level classification. For the Ultralytics classification format, the images are arranged in train, val and test folders with one sub-folder per class, using an 80:10:10 split stratified by class so that every class appears in all three subsets. The test set is kept unseen until the final evaluation. PlantVillage is the only dataset used in this work. Table III summarises the dataset and its split.

TABLE III SUMMARY OF THE PLANTVILLAGE DATASET AND ITS SPLIT

Property	Value
Source	PlantVillage open-access leaf image repository [6]
Total images	54,306
Crop species	14
Diseases	26
Classes (crop and condition, including healthy)	38
Image type	RGB photographs of single leaves
Split (train / val / test)	80 : 10 : 10, stratified by class (approx. 43,445 / 5,431 / 5,430 images)
Folder format	One sub-folder per class inside train, val and test

.

B. Training Configuration

The model is created from the pre-trained weights yolo26n-cls.pt and fine-tuned with the Ultralytics training interface, with the image size set to 224, a batch size chosen to fit the available memory (for example 64) and a training budget of about 50 epochs with early stopping based on validation accuracy. The optimiser and learning-rate schedule follow the framework defaults, and the validation split is used to select the best checkpoint (best.pt). After training, the best model is evaluated on the unseen test split and exported to ONNX for efficient inference in the back end of the web application [13]. These hyper-parameter values are starting points and should be updated to match the actual training run.

C. Prediction And Result Interpretation

For a new image the model outputs a probability for each class. The class with the highest probability is shown as the possible disease name, and its probability is displayed as the confidence value. The next most probable classes are listed as possible alternatives, which communicates that similar-looking symptoms can have different causes. A severity label (for example, mild, moderate or severe) is derived from the prediction and affected-area information, and the matching cause, treatment and prevention content is retrieved. When confidence is low, or severity is high, the interface advises the user to consult a local agricultural expert.

D. Evaluation Metrics

Model performance is measured on the test set using top-1 accuracy, top-5 accuracy, precision, recall and F1-score, supported by a confusion matrix to reveal which diseases are confused with each other. Class activation visualisation methods such as Grad-CAM [15] can be used to check that the network focuses on the lesion area rather than on the background.

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

III. Implementation

AgriScan AI is implemented as a responsive web application so that it can be used from a smartphone browser without installation. The prototype interface was designed for non-technical users, with a left-hand navigation menu, large action buttons and short guidance text. The model can be built and trained with Python and the Ultralytics framework (PyTorch) and, after export to ONNX, served through an API endpoint that the web interface calls after an image is submitted.

A. Application Modules

The main modules of the prototype are listed in Table IV.

Table Iv

Main Modules Of The Agriscan Ai Prototype.

Module	Description
Overview	Dashboard with greeting, location, weather summary, active fields, scans this month, healthy-crop percentage and next check-in reminder
Scan a crop	Upload or camera capture, crop selection (auto-detect or manual), field selection, sample images and scanning tips
Scan history	Record of earlier scans and results for review
My fields	Management of fields and the crops grown in each
Krishi Mitra	Conversational assistant for farming questions and result explanation
Outbreak map	Awareness of disease reports in the surrounding region
Farm diary	Notes and activity records for the farm
Shops & mandi	Access to input shops and market information

1.User Interface Walkthrough

Fig. 1 shows the Overview screen. After login the farmer sees a personalised greeting, the current location, a local weather card (temperature, humidity, rain chance and wind) and summary tiles for active fields, scans this month, the percentage of healthy crops and the next scheduled check-in. Prominent buttons lead directly to leaf scanning and to the Krishi Mitra assistant.

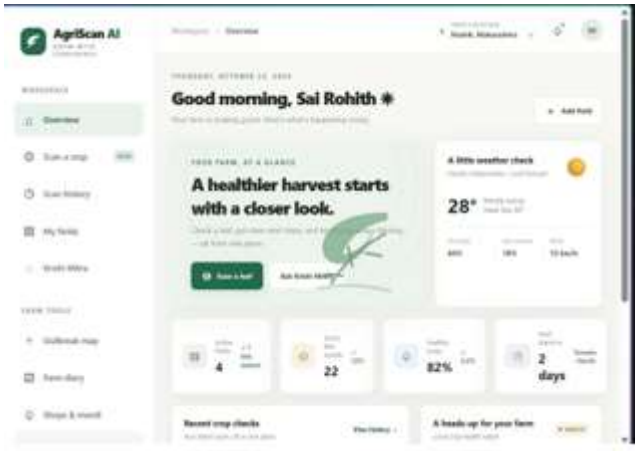


Fig. 1. AgriScan AI Overview dashboard with weather, field summary and

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

Fig. 1. AgriScan AI Overview dashboard with weather, field summary and quick actions.

Fig. 2 shows the Scan a crop screen. The user can choose a photo from the device or use the camera, select the crop being grown (or let the system auto-detect it) and attach the scan to a field. Three short tips encourage good image capture: find a well-lit spot, get close and keep the image sharp, and show both a close-up and a wider plant photo for context. Sample images of tomato, cotton and grape leaves allow users to explore the application without their own photos.

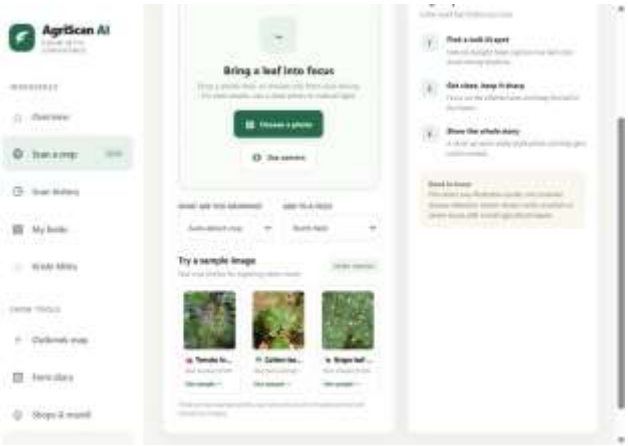


Fig. 2. Scan a crop screen: photo upload or camera, crop and field selection sample images and scanning tips.

Fig. 3 shows the result screen. The result card displays the crop and the possible disease name, a status badge, the confidence bar, possible alternative conditions and a severity level. Tabs present the cause, treatment and prevention information and allow the user to ask the assistant. Action buttons let the user listen to the result, share it, ask Krishi Mitra or request expert help. The interface clearly states that the current results are illustrative and that uncertain or severe issues should be verified with a local agricultural expert.

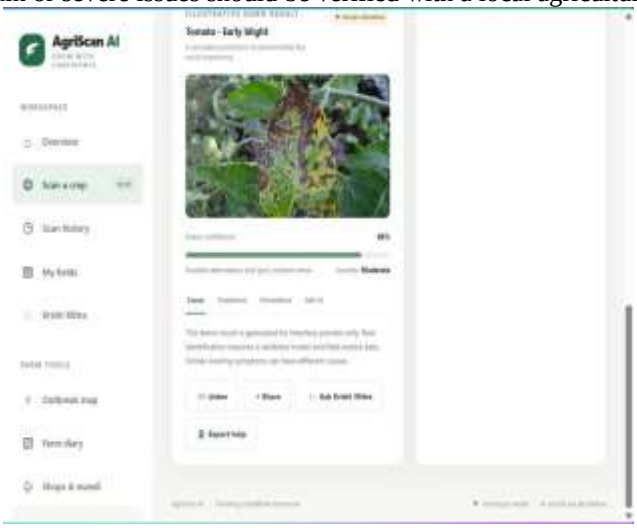


Fig. 3. Result screen showing disease name, confidence, severity, cause/treatment/prevention tabs and action buttons (illustrative demo result).

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

A. Security, Privacy and Responsible Use

The application should authenticate users before storing field data, validate uploaded files by type and size, and avoid exposing personal or location information beyond what is needed. Because an incorrect diagnosis can lead to unnecessary or harmful treatment, the system displays confidence and alternatives rather than a single unqualified answer, and it recommends expert confirmation for uncertain or severe cases.

1V. Results And Discussion

The three screenshots in Figs. 1–3 are taken from the working AgriScan AI prototype. In its present state the prototype runs in demonstration mode: its sample scans return simulated results intended to demonstrate the user experience, and the interface states that real identification requires a validated model and field-tested data. The performance values in Table V are therefore provided as illustrative expected values for a YOLO26n-cls model fine-tuned on the PlantVillage dataset described in Section IV. They are not claimed as verified measurements and must be replaced with the team's measured results from the trained model before final submission.

Table V

Illustrative Prototype Evaluation Metrics For Agriscan Ai (Yolo26n-Cls, Plantvillage Test Split).

Metric	Illustrative Value	Evaluation Basis
Top-1 accuracy	96%	Illustrative PlantVillage test split
Top-5 accuracy	99.6%	Illustrative PlantVillage test split
Precision (macro)	95.5%	Illustrative PlantVillage test split
Recall (macro)	95%	Illustrative PlantVillage test split
F1-score (macro)	95.2%	Harmonic mean of illustrative precision and recall
Average prediction time	1.2 s	Illustrative upload-to-result time
Usability rating	4.5 / 5	Illustrative assessment with a small group of users

A. End-To-End Scenario

A representative scenario begins with a farmer in the field noticing dark, yellow-bordered spots on a tomato leaf. The farmer opens Scan a crop, takes a photo with the camera, leaves the crop as auto-detect and assigns the scan to the North field. The image is pre-processed and passed to the YOLO26n model, which returns the most probable class, in the illustrated example Tomato · Early blight, with 86% confidence, moderate severity and leaf spot and nutrient stress as alternatives. The farmer reads the cause, treatment and prevention tabs, listens to the explanation in audio form if reading is inconvenient, and, because severity is moderate, asks Krishi Mitra a follow-up question or requests expert help. The scan is saved in the history of the field so that the next check-in can show whether the condition is improving.

B. Discussion

The choice of YOLO26n reflects the deployment setting. Rural users may depend on low-cost servers and limited connectivity, so a model with about 2.8 million parameters and roughly half a GFLOP per 224-pixel image keeps response time and hosting cost low, and ONNX export leaves open a later move to on-device or edge inference [13]. Larger

International Journal for Research Trends in Social Science & Humanities
Volume 4, Issue 5
ISSN: 2584-2455
Sep-Oct-2026, PP 1-11

YOLO26 variants (s, m, l and x) can be substituted if accuracy matters more than speed, with no change to the rest of the application.

The principal contribution of AgriScan AI is accessibility: a trained YOLO26n model is wrapped in an interface that a farmer can use without technical knowledge, and the output includes the information needed to act on a diagnosis. Compared with manual inspection, the system gives a consistent first assessment in seconds and reduces the number of situations in which an expert visit is needed merely to identify a common disease. Displaying confidence, alternatives and severity helps users interpret the prediction correctly, while the expert-help option provides a safety net.

Published studies report very high accuracy on laboratory-style images but lower accuracy on field photographs with cluttered backgrounds [2], [4], [8]. Achieving reliable performance in real farms will therefore require training with locally collected images and testing under varied lighting and backgrounds, as listed in the future scope.

1. Advantages And Limitations

A. Advantages

- Fast analysis of a leaf image without waiting for an expert visit.
- Reduces dependence on manual, error-prone visual inspection.
- Accessible from a smartphone browser, including in remote areas with basic connectivity.
- Average prediction time

2. S Illustrative Upload-To-Result Time

- Provides disease name, confidence, severity and treatment and prevention guidance in one place.
- Usability rating 4.5 / 5 Illustrative assessment with a
- small group of users
- Scan history and field management support monitoring of crop health over time.
- Simple interface with sample images and scanning tips designed for non-technical users.
- Modular architecture allows new crops, diseases and languages to be added by extending the model and content.

A. Limitations

- Prediction quality depends on image quality, lighting, focus and the visible stage of the disease.
- Models trained on laboratory images may perform worse on field photographs with complex backgrounds.
- Only the crops and diseases present in the training data can be recognised; unseen diseases may be misclassified.
- Different diseases, pest damage and nutrient deficiencies can look similar, so the result is a possible diagnosis rather than a certain one.
- The current prototype uses simulated results; a validated model and field testing are still required.
- Internet connectivity is needed for server-based inference.
- Ultralytics YOLO26 is distributed under the AGPL-3.0 licence, with an enterprise licence available for commercial use, so licensing should be reviewed before any commercial deployment [13].

3. Future Scope

Future work will first replace the demonstration results with a fully trained and evaluated YOLO26n model, followed by field testing with farmers. Further extensions include support for more crops and diseases, collection of region-specific image datasets, regional language and voice support, and an offline mode that runs a compressed model on the device for areas with poor connectivity. Additional improvements include severity estimation by lesion segmentation, explainable predictions using heat maps, integration of weather data for disease-risk forecasting, alerts based on the outbreak map, and direct connection to agricultural experts and local input suppliers.

V. Conclusion

This paper presented AgriScan AI, a web-based application that uses the lightweight YOLO26n model, a CNN-based network fine-tuned on the PlantVillage dataset, to detect plant leaf diseases from a photograph. Farmers can upload or capture an image of an affected leaf, learn whether the plant is healthy or diseased, see the possible disease name with a confidence value, and receive guidance on cause, treatment and prevention through a simple interface. By providing a quick first assessment, the system reduces the need for manual inspection and frequent expert visits and makes disease detection more accessible in remote areas.

The prototype demonstrates the complete user experience, including scanning, results, scan history, field management and assistance. The next step is to train and validate YOLO26n on the PlantVillage dataset and on field images so that the illustrative values reported here can be replaced with measured results. With these improvements, AgriScan AI can become a practical tool for early disease detection and healthier harvests.

References

1. S. Savary, L. Willocquet, S. J. Pethybridge, P. Esker, N. McRoberts, and A. Nelson, “The global burden of pathogens and pests on major food crops,” Nature Ecology & Evolution, vol. 3, pp. 430–439, 2019.
2. J. G. A. Barbedo, “A review on the main challenges in automatic plant disease identification based on visible range images,” Biosystems Engineering, vol. 144, pp. 52–60, 2016.
3. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, 2015.
4. S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” Frontiers in Plant Science, vol. 7, Art. no. 1419, 2016.
5. K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.
6. 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.
7. 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.
8. J. G. A. Barbedo, “Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification,” Computers and Electronics in Agriculture, vol. 153, pp. 46–53, 2018.

9. A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.
10. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.
11. S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Trans. Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.
12. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.
13. Ultralytics, “Ultralytics YOLO26 documentation: Image classification,” 2026. [Online]. Available: https://docs.ultralytics.com/tasks/classify/ (accessed Nov. 3, 2026).
14. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.
15. R. R. Selvaraju et al., “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Computer Vision (ICCV), 2017, pp. 618–626.