

KrishiNethra-RRSP: An Integrated Smart Agriculture Platform Using AI, Deep Learning and IOT

Pramith Rao¹, Rolvin Dsouza², Ranjith³, Sadananda SS⁴, Prof. Shruthi⁵, Assistant Professor
^{1,2,3,4,5}*Department of Computer Science and Engineering, Shree Devi Institute of Technology Visvesvaraya Technological University*

Abstract—Farmers often depend on separate sources for crop-health information, irrigation support, market-price updates, agricultural equipment, government schemes, and selling opportunities. This fragmentation can increase the effort required to make timely decisions, especially when digital services are difficult to access or are not available in a preferred language. This paper presents KrishiNethra-RRSP, an integrated smart agriculture platform that combines artificial intelligence, machine learning, deep learning, and Internet of Things technologies with farmer-oriented digital services. The proposed platform includes image-based crop disease classification, IoT sensor integration for field monitoring and irrigation support, market and scheme information, equipment access, farmer-producer-organisation support, and multilingual voice assistance. The disease-classification component uses a Keras model with a 224×224 image input and twenty configured output classes covering black pepper, corn, rice, sugarcane, wheat, and an unknown category. The supplied model configuration records a test accuracy of 94.75% and uses a confidence threshold of 0.80. These values are reported as project-level records and are not treated as evidence of field performance. The paper describes the system architecture, implementation approach, available evaluation evidence, limitations, and future work required for reproducible and field-oriented validation.

Index Terms—Smart agriculture, artificial intelligence, machine learning, deep learning, Internet of Things, crop disease detection, convolutional neural network, precision irrigation.

I. INTRODUCTION

Agriculture depends on timely decisions related to crop selection, crop health, irrigation, weather conditions, input usage, market prices, and access to public support schemes. In many situations, these

activities are handled through different applications, local contacts, manual records, or disconnected online services. Such fragmentation may increase the workload of farmers and make important information difficult to locate. Language barriers, limited digital literacy, unreliable connectivity, and dependence on intermediaries can further reduce the usefulness of digital agricultural services.

Artificial intelligence (AI), machine learning (ML), deep learning (DL), and the Internet of Things (IoT) provide complementary methods for addressing some of these challenges. Image-based deep learning can support the identification of visible crop symptoms, while IoT sensors can collect field parameters such as soil moisture and environmental conditions. Machine learning can also be used for agricultural recommendation and price-analysis tasks when suitable data are available. However, individual technical solutions do not automatically provide a complete farmer-oriented service.

The information must be connected through a usable interface, secure backend services, and clear decision-support workflows. KrishiNethra-RRSP is designed as an integrated platform for this purpose. The project brings together crop disease diagnosis, IoT-based field monitoring, irrigation support, market information, agricultural equipment access, government-scheme information, farmer-producer-organisation (FPO) support, and multilingual interaction. The platform follows a modular design so that the sensing, backend, inference, and user-interface components can be developed and evaluated separately.

The main contributions of this paper are:

- an integrated architecture for combining agricultural advisory and farmer-service modules;
- an IoT-oriented field-monitoring layer intended to

- connect sensors, a microcontroller, communication services, and application logic;
- an image-classification workflow based on a Keras model and confidence-aware responses;
- a farmer-oriented interface that considers multilingual interaction and voice assistance; and
- a transparent account of the currently available evaluation evidence and the limitations that must be addressed before field deployment.

II. LITERATURE REVIEW

A. Artificial Intelligence and Machine Learning in Agriculture

Machine learning has been applied to crop recommendation, yield estimation, agricultural commodity-price analysis, and resource planning. The effectiveness of these systems depends on the quality, coverage, and representativeness of the input data. Crop recommendation may use soil, climate, and geographical variables, whereas price forecasting generally requires historical records and consideration of seasonal and external factors. These observations influenced the design of KrishiNethra-RRSP, where recommendation and market services are treated as data-dependent modules rather than as outputs that can be assumed to be accurate for every location.

B. Deep Learning for Plant Disease Recognition

Mohanty *et al.* demonstrated the use of deep convolutional neural networks for image-based plant disease recognition and also highlighted the difference between controlled images and images captured in real agricultural environments [?]. This distinction is important because field images may contain shadows, complex backgrounds, blur, multiple leaves, and symptoms that are not represented in a training dataset.

Convolutional neural networks (CNNs) learn visual features directly from images and have been widely used for plant disease classification. MobileNet introduced depthwise separable convolutions as a way to reduce computation and model size, making the architecture relevant to resource-constrained applications [?]. Recent reviews also discuss the use of transfer learning, lightweight networks, explainable AI, and improved datasets for plant disease detection [12], [13]. These studies support the need to evaluate not only accuracy but also

model generalisation, computational requirements, and practical usability.

C. Internet of Things in Smart Farming

IoT-based agricultural systems generally contain sensors, a microcontroller, a communication mechanism, a backend service, and an actuator or dashboard. Soil-moisture sensors can provide information for irrigation decisions, while temperature and humidity sensors can support environmental monitoring. An overview of IoT-based smart irrigation systems describes the role of sensors and communication technologies in precision agriculture [1]. IoT and deep learning can also be combined for crop monitoring and disease-related tasks, although practical deployments must consider connectivity, energy consumption, security, sensor errors, and hardware cost [2], [12].

D. Research Gap

The reviewed studies commonly focus on one major problem, such as disease classification, irrigation, crop recommendation, or commodity-price forecasting. KrishiNethra-RRSP addresses the system-integration gap by placing multiple farmer services behind a common web platform. The project does not claim that all modules have identical maturity or validation status. Instead, it separates implemented components, proposed service workflows, and evaluation requirements so that future development can be measured more clearly.

III. METHODOLOGY

The proposed system is organised into four main layers: the field and sensing layer, the application and data layer, the AI/ML inference layer, and the farmer interaction layer.

A. Field Sensing and IoT Layer

The IoT layer is intended to collect field information through a microcontroller-based sensing unit. An ESP32 can act as the local controller, while soil-moisture and environmental sensors provide readings for monitoring and irrigation-related decisions. The sensing unit communicates with an application service through a communication bridge or network request. The backend can store accepted readings, display recent values, and provide the status required

by the dashboard.

The irrigation logic is designed around explicit conditions rather than an unexplained prediction. Let m represent the measured soil-moisture value, T_{dry} represent the configured dry-soil threshold, T_{wet} represent the configured upper threshold, and r represent the rain-detection state. A simplified control policy is expressed as:

Pump ON if $m \leq T_{dry} \wedge r = 0$, (1)

Pump OFF if $m \geq T_{wet} \vee r = 1$. (2)

The threshold values must be calibrated for the sensor, soil type, crop, and irrigation arrangement. The controller should also include manual override, safe default behaviour, and protection against repeated or conflicting commands.

B. AI, ML and Deep Learning Layer

The disease-diagnosis workflow accepts a crop image and applies a fixed preprocessing procedure before inference. The supplied predictor converts the image to RGB format, resizes it to 224×224 pixels, and sends it to the Keras model. The predicted class is selected using the maximum output score. The configured classes cover black pepper, corn, rice, sugarcane, wheat, and an unknown category.

The response logic is confidence-aware. If the predicted class is “Unknown”, the system returns an invalid-image response. If the confidence is below the configured threshold, the system returns an uncertain response and requests a clearer image. For accepted predictions, the response includes the crop, disease label, confidence value, and a confidence status. This approach is intended to reduce overconfident responses, but a neural-network score should not automatically be interpreted as a calibrated probability or as a substitute for expert diagnosis.

The general classification process can be written as:

$$\hat{y} = \arg \max_i f_{\theta}(x)_i, \quad (3)$$

where x is the preprocessed image, f_{θ} is the trained model, and $\hat{y}$ is the predicted class. The confidence-aware response is then determined by the predicted score and the configured threshold.

C. Farmer-Service Layer

The platform is designed to bring several farmer-oriented services into one interface. These services include market-price information, government-scheme guidance, agricultural equipment access,

FPO-based selling support, and multilingual voice assistance. Their usefulness depends on reliable external data, regular updates, location-specific information, and clear explanations. Therefore, the platform should present these services as decision support and should identify the source and update time of information whenever possible.

IV. IMPLEMENTATION AND SYSTEM ENGINEERING

The architecture follows a modular edge-to-application pattern. The IoT device forms the field layer. A communication bridge or network endpoint transfers sensor readings to the backend. The Django and Django REST Framework layer handles authentication, business rules, API requests, and persistence. PostgreSQL is used as the intended relational data layer. The AI inference component is kept separate from the main business logic so that the model can be updated without redesigning the complete application.

TABLE I MAIN TECHNOLOGY COMPONENTS

Layer	Technology or responsibility
Field layer	ESP32 and agricultural sensors
Communication	Serial or network-based data transfer
Backend	Django and Django REST Framework
Database	PostgreSQL
Authentication	JWT-based authentication
AI inference	TensorFlow/Keras model and Python predictor
Image processing	RGB conversion and 224×224 resizing
Frontend	HTML5, CSS3, and modular JavaScript
Farmer services	Market, schemes, equipment, FPO and voice-support workflows

The main data flow is as follows:

- 1) Sensors collect field readings through the microcontroller.
- 2) The communication layer transfers readings to the backend.
- 3) The backend validates and stores the received information.
- 4) The dashboard displays sensor values, device status, and available service information.
- 5) A user-uploaded image is preprocessed and passed to the disease-classification model.
- 6) The prediction service returns a disease label or an uncertain/invalid response based on the configured decision rules.

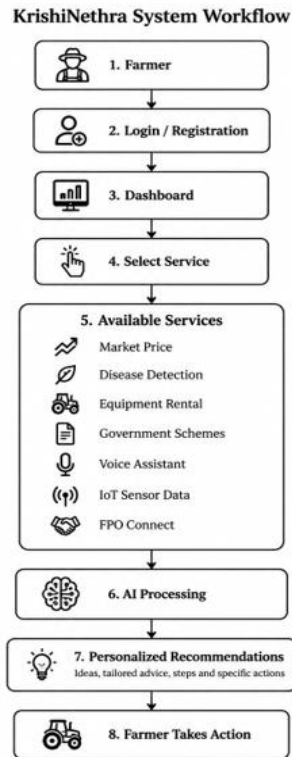


Fig. 1. KrishiNethra System Workflow.

V. IMPLEMENTATION

The available disease-model archive contains four principal files: `model.keras`, `class_names.json`, `thresholds.json`, and `predictor.py`. The predictor loads the model and configuration files, reads the class names, performs image preprocessing, and generates a response according to the confidence rules. The class configuration contains twenty classes, including an unknown category.

The project-level technology plan uses HTML5, CSS3, and modular JavaScript for the frontend, Django with Django REST Framework for backend services, PostgreSQL for structured data storage, and JWT-based authentication. The IoT module is designed to exchange sensor information with the backend, while the dashboard provides a common access point for the different service modules.

The supplied configuration records a model version of v1, a confidence threshold of 0.80, and a test accuracy value of 0.9474781155586243. The original training dataset, training notebook, class distribution, split strategy, optimizer, epoch count, and complete training history were not present in the supplied

model archive. These details are therefore not invented in this paper and should be added after the original training records are recovered.

VI. RESULTS AND EVALUATION

A. Available Model Result

The supplied configuration records a test accuracy of 0.9474781155586243, which is approximately 94.75% after rounding. This is reported as a recorded project metric. The archive does not include the test predictions, confusion matrix, class-wise precision, recall, F1-score, or the number of samples used for testing. Consequently, the result cannot establish that all classes perform equally well or that the model will achieve the same performance on field images.

B. IoT and System-Level Evaluation

The IoT and application modules require evaluation beyond successful code execution. Important measurements include sensor accuracy, communication reliability, command latency, pump response, power consumption, behaviour during network failure, and recovery after reconnection. Test cases should cover dry-soil readings, adequate-moisture readings, rain detection, sensor noise, manual override, and conflicting commands. No measured field dataset, long-duration pump test, water-consumption comparison, or latency distribution is claimed here because those records were not available in the supplied materials.

C. Evaluation Limitations

The current evaluation is limited by the evidence available in the model archive and project presentation. A stronger study should include a documented dataset, class-wise metrics, a confusion matrix, confidence calibration, images from unseen plants, and an external field-image test set. For the IoT layer, experiments should be conducted under different soil conditions, weather conditions, connectivity states, and sensor-error scenarios. User testing with farmers or agricultural practitioners is also necessary to assess usability and comprehension.

D. Discussion

KrishiNethra-RRSP is intended to reduce the need to use separate tools for different agricultural activities. Its main contribution is the integration of AI-based

crop disease support, IoT-oriented monitoring, and farmer-service workflows within one platform. The modular architecture allows the disease model, sensor service, backend, and frontend to be improved independently.

The recorded model accuracy provides an initial project-level indicator, but it should be interpreted together with the missing dataset and class-wise evaluation details. The unknown category and uncertain response can help the system avoid presenting every input as a reliable diagnosis. Nevertheless, image-based predictions should be treated as preliminary decision support. Farmers should be encouraged to consult agricultural experts when the prediction is uncertain or when the crop condition is serious.

The IoT component also introduces safety and operational concerns. Sensor drift, communication loss, incorrect threshold configuration, and actuator failure may affect irrigation decisions. Hardware protections, logging, manual emergency control, and safe fallback states should be included before the system is used in a real farm. Authentication data, sensor records, and farmer information should be protected through suitable access-control and security practices.

VII. CONCLUSION

This paper presented KrishiNethra-RRSP, an integrated smart agriculture platform that combines AI, machine learning, deep learning, IoT-oriented field monitoring, and farmer-focused services. The proposed architecture uses an ESP32-based sensing layer, a Django and Django REST Framework backend, PostgreSQL, JWT-based authentication, and a Keras image-classification component. The supplied model configuration records a test accuracy of approximately 94.75%, uses a 224×224 input image, and applies a confidence threshold of 0.80 across twenty configured classes. These values are reported transparently, while the absence of the original dataset, class-wise metrics, and field measurements is acknowledged. The platform provides a foundation for further development, but additional technical testing and real-world validation are required before making stronger claims about reliability, safety, and agricultural impact.

ACKNOWLEDGMENT

The authors thank Ms. Shruthi, project guide, and the faculty members of the Department of Computer Science and Engineering, Shree Devi Institute of Technology, Mangaluru, for their guidance and support during the development of the project.

REFERENCES

- [1] L. García *et al.*, “IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture,” *Sensors*, vol. 20, no. 4, Art. no. 1042, 2020, doi: 10.3390/s20041042.
- [2] C.-J. Chen *et al.*, “An AIoT based smart agricultural system for pests detection,” *IEEE Access*, vol. 8, pp. 180750–180761, 2020, doi: 10.1109/ACCESS.2020.3024891.
- [3] A. Rehman, M. S. Khan, M. S. S. Khan, Z. Ullah, and M. A. Khan, “A novel deep learning method for detection and classification of plant diseases,” *Complex & Intelligent Systems*, vol. 8, pp. 507–524, 2022, doi: 10.1007/s40747-021-00536-1.
- [4] S. Janarthan, S. Thuseethan, S. Rajasegarar, and J. Yearwood, “P2OP–Plant Pathology on Palms: A deep learning-based mobile solution for in-field plant disease detection,” *Computers and Electronics in Agriculture*, vol. 202, Art. no. 107371, 2022, doi: 10.1016/j.compag.2022.107371.
- [5] W. Chen, J. Chen, R. Duan, Y. Fang, Q. Ruan, and D. Zhang, “MS-DNet: A mobile neural network for plant disease identification,” *Computers and Electronics in Agriculture*, vol. 199, Art. no. 107175, 2022, doi: 10.1016/j.compag.2022.107175.
- [6] A. Rehman *et al.*, “The power of transfer learning in agricultural applications: AgriNet,” *Frontiers in Plant Science*, vol. 13, 2022, doi: 10.3389/fpls.2022.992700.
- [7] J. Lu, R. Shi, J. Tong, W. Cheng, X. Ma, and X. Liu, “Lightweight Method for Plant Disease Identification Using Deep Learning,” *Intelligent Automation & Soft Computing*, vol. 37, no. 1, pp. 525–544, 2023, doi: 10.32604/iasc.2023.038287.
- [8] Y. Zhong, Z. Teng, and M. Tong, “LightMixer:

- A novel lightweight convolutional neural network for tomato disease detection,” *Frontiers in Plant Science*, vol. 14, 2023, doi: 10.3389/fpls.2023.1166296.
- [9] H. Guan, C. Fu, G. Zhang, K. Li, P. Wang, and Z. Zhu, “A lightweight model for efficient identification of plant diseases and pests based on deep learning,” *Frontiers in Plant Science*, vol. 14, Art. no. 1227011, 2023, doi: 10.3389/fpls.2023.1227011.
- [10] “PMVT: A lightweight vision transformer for plant disease identification on mobile devices,” *Frontiers in Plant Science*, vol. 14, 2023, doi: 10.3389/fpls.2023.1256773.
- [11] “A robust and light-weight transfer learning-based architecture for accurate detection of leaf diseases across multiple plants using less amount of images,” *Frontiers in Plant Science*, 2023, doi: 10.3389/fpls.2023.1321877.
- [12] V. S. Dhaka *et al.*, “Role of Internet of Things and deep learning techniques in plant disease detection and classification: A focused review,” *Sensors*, vol. 23, no. 18, Art. no. 7877, 2023, doi: 10.3390/s23187877.
- [13] “Plant leaf disease detection, classification, and diagnosis using computer vision and artificial intelligence: A review,” *IEEE Access*, vol. 12, pp. 37443–37469, 2024, doi: 10.1109/ACCESS.2024.3373001.
- [14] “IoT -based plant disease detection using machine learning: A systematic literature review,” in *Proc. 2024 Int. Conf. on Smart Applications, Communications and Networking*, 2024, doi: 10.1109/SmartNets61466.2024.10577751.
- [15] “Pre-harvest to post-harvest: A review of AI and IoT applications in smart agriculture and the prospects of 6G-enabled IoT framework,” in *Proc. 2024 27th Int. Symp. on Wireless Personal Multimedia Communications*, 2024, doi: 10.1109/WPMC63271.2024.10863521.