



AI-enabled crop yield detection and classification using ML and DL

Prof. Ullas H S^a, Chethana S^a, Chinmayee P V^a, Hemamrutha M P^a, and Janhavi H^a

^aDepartment of Computer Science and Engineering, Kalpataru Institute of Technology, NH-206, Tiptur – 572201, Karnataka, India

ABSTRACT :

Agricultural decisions, including crop selection, yield estimation, disease identification, and fertilizer planning, depend on crop, soil, and environmental conditions. This paper presents AgriPredict, a web-based agricultural advisory system integrating machine learning (ML), deep learning (DL), and rule-based methods. The system estimates crop yield from crop type, rainfall, pesticide usage, and average temperature. It also uses MobileNetV2 for crop-image classification and VGG16 for detecting supported plant-disease classes. Additional functions include disease-risk assessment, fertilizer and crop recommendations, weather information, user authentication, prediction history, a discussion forum, and a marketplace. The application was developed using Python, Flask, TensorFlow or Keras, scikit-learn, and SQLite. Unit and integration testing covered the system's main functions and input handling. The report states that tested modules produced expected outputs for the inputs used; however, quantitative model-accuracy and response-time measurements are not reported. AgriPredict is intended for academic demonstration and preliminary agricultural guidance. Its outputs depend on input quality, supported model classes, and implemented recommendation rules, and should be verified using local expertise and soil-test results.

Keywords: Smart agriculture; crop yield prediction; machine learning; deep learning; plant disease detection; crop classification; agricultural advisory system

1. Introduction

Agricultural decisions are influenced by factors such as crop type, soil properties, rainfall, temperature, and plant health. Farmers consider these conditions when selecting crops, estimating yield, identifying possible diseases, and planning fertilizer use. Previous studies have applied machine-learning methods to crop recommendation and yield prediction (Mahale et al., 2024; Manideep & Kharb, 2022; Nishant et al., 2020). Deep-learning approaches have also been used for image-based plant-disease detection and classification (Bala et al., 2025; Paymode & Malode, 2022).

This paper presents AgriPredict, a web-based agricultural advisory system that integrates several support functions in one application. Its crop-yield module estimates yield using crop type, rainfall, pesticide usage, and average temperature. The image-based modules use MobileNetV2 for crop classification and VGG16 for plant-disease detection. The system also includes rule-based services for disease-risk assessment, fertilizer recommendation, and crop suitability recommendation.

AgriPredict is implemented using Python and Flask, with TensorFlow or Keras and scikit-learn for model integration and SQLite for data storage. It also provides user authentication, prediction history, weather information, a discussion forum, and a marketplace. The system is intended to provide preliminary agricultural guidance; its results depend on input quality, the classes represented in the trained models, and the rules used by its recommendation modules.

The Flask application receives user inputs and directs them to the appropriate prediction or recommendation service. Numerical data is processed by the yield model or relevant rule-based module. Uploaded images are validated and preprocessed before being analysed by the selected deep-learning model. The system displays predictions and recommendations through its web interface. Authenticated users can save results in their prediction history for later reference. Functional and integration tests were conducted on the main modules using valid and invalid inputs to verify expected outputs and error handling. The platform integrates these services while presenting results as preliminary guidance rather than a substitute for expert advice. Image-based modules are limited to the crop and disease classes represented in their training datasets. AgriPredict is designed as an academic prototype for small-scale use, and its outputs should be interpreted with these limitations in mind. Fertilizer and crop recommendations should be verified using local agricultural expertise and soil-test results. Further evaluation using larger and more diverse datasets could help assess the reliability of the prediction models. Future enhancements may include improved model accuracy and validation under different agricultural conditions. The system could also be extended to support additional crop varieties and plant diseases. Mobile-friendly access could make the platform more convenient for farmers. The modular architecture of AgriPredict allows individual services to be maintained and improved independently. The integration of machine-learning and deep-learning techniques provides different forms of agricultural decision support within a single platform. However, the practical effectiveness of the system requires further assessment using real-world agricultural data and feedback from farmers. The proposed system provides a foundation for developing more comprehensive and data-driven agricultural advisory solutions. With further testing and refinement, AgriPredict could contribute to more informed crop management and sustainable agricultural practices.

2. Related Work

Machine-learning and deep-learning methods have been applied to agricultural problems such as crop recommendation, yield prediction, and plant-disease identification. These studies provide background for AgriPredict's numerical prediction and image-analysis functions. Table 1 summarizes five studies reviewed in the project report.

Table 1 - Summary of selected agricultural studies

Study	Method or focus	Contribution and limitation noted in the project review
Bala et al. (2025)	Lightweight CNN using MobileNetV2	Applied MobileNetV2 to plant-leaf disease detection; further testing with diverse field images and environmental conditions is needed.
Mahale et al. (2024)	Machine learning and LSTM	Recommended crops for Maharashtra using agricultural and weather data; does not integrate yield prediction and image-based services.
Manideep and Kharb (2022)	Regression, decision tree, random forest, SVM, and gradient boosting	Compared methods for crop-yield prediction; does not provide an integrated agricultural platform.
Paymode and Malode (2022)	CNN with VGG transfer learning	Classified multi-crop leaf diseases; results may be affected by image quality, lighting, background, and dataset diversity.
Nishant et al. (2020)	Machine learning and stacking regression	Predicted yield using Indian state, district, season, cultivated-area, and crop data; focuses on yield rather than image-based services.

Crop-recommendation and yield-prediction research uses agricultural, environmental, and historical data to estimate suitable crops or expected production. Mahale et al. (2024) studied crop recommendation and forecasting for Maharashtra, while Ahamed et al. (2015) examined data-mining techniques to predict crop yields and recommend crops in Bangladesh. These studies illustrate how agricultural data can support more than one planning decision, while also reflecting the importance of regional context when designing recommendations.

Yield-prediction studies address the related task of estimating expected production. Nishant et al. (2020) used Indian agricultural information, including state, district, season, cultivated area, and crop. Manideep and Kharb (2022) compared several machine-learning algorithms for crop-yield prediction. Nevavuori et al. (2019) investigated deep convolutional neural networks for yield prediction. The varied inputs and methods in these studies show that yield estimates are closely tied to the data and modelling approach used. Therefore, an evaluation should explain the dataset, input variables, and metrics rather than report a prediction alone.

Random forests and sequence models are also part of the broader methodological background. Breiman (2001) described random forests as an ensemble learning method, while Hochreiter and Schmidhuber (1997) introduced long short-term memory networks for sequence learning. These references are relevant to agricultural studies that use tree-based methods or time-dependent data. They provide methodological context, but do not by themselves establish the performance of the specific models used in AgriPredict.

Image-based plant analysis is another relevant area. Bala et al. (2025) studied MobileNetV2 for plant-leaf disease detection, and Paymode and Malode (2022) applied VGG-based transfer learning for multi-crop disease classification. Ferentinos (2018), Mohanty et al. (2016), and Sladojević et al. (2016) also examined deep-learning approaches for identifying plant diseases from images. Fuentes et al. (2017) investigated deep-learning-based recognition of tomato diseases and pests. Together, these studies provide context for using image classifiers to analyse visual crop symptoms.

The results of image-based systems depend on what the model has seen during training. A model trained on a limited set of crops or disease categories may not correctly handle an unfamiliar class. Differences in lighting, background, image framing, and image quality can also affect predictions. For this reason, the report describes AgriPredict's image modules as limited to the crop and disease classes in their training datasets, and recommends treating their outputs as preliminary guidance.

The architectures used in AgriPredict have foundations in computer-vision research. Simonyan and Zisserman (2015) described very deep convolutional networks for image recognition, providing background for VGG-based models. Sandler et al. (2018) introduced MobileNetV2, a lightweight architecture. Krizhevsky et al. (2012) demonstrated deep convolutional networks for large-scale image classification, contributing to the development of modern image-analysis methods. These works provide architectural context; they should not be interpreted as direct evaluations of AgriPredict's trained models. Machine-learning frameworks support the implementation of these methods. Abadi et al. (2016) described TensorFlow as a system for large-scale machine learning, and Pedregosa et al. (2011) introduced scikit-learn as a Python machine-learning library. In the project, TensorFlow/Keras supports deep-learning model integration, while scikit-learn is used in the crop-yield module. The report also describes handling compatibility requirements between the VGG16 and MobileNetV2 model environments.

The reviewed studies address important agricultural problems, but they generally focus on individual prediction or classification tasks. AgriPredict brings crop-yield prediction, crop-image classification, plant-disease detection, disease-risk assessment, and crop and fertilizer recommendations together in one web application. It also includes user authentication, prediction history, and community features. This integration is the project's system-level focus. The application is intended to provide preliminary guidance; its results depend on input quality, trained-model classes, and the recommendation rules implemented. The unified platform provides users with access to multiple agricultural support functions through a common interface.

3. Methodology and System Design

AgriPredict is a web-based agricultural advisory system that combines machine-learning prediction, deep-learning image analysis, and rule-based recommendations. Its architecture separates the user interface, application logic, prediction services, and data storage.

3.1. System Architecture

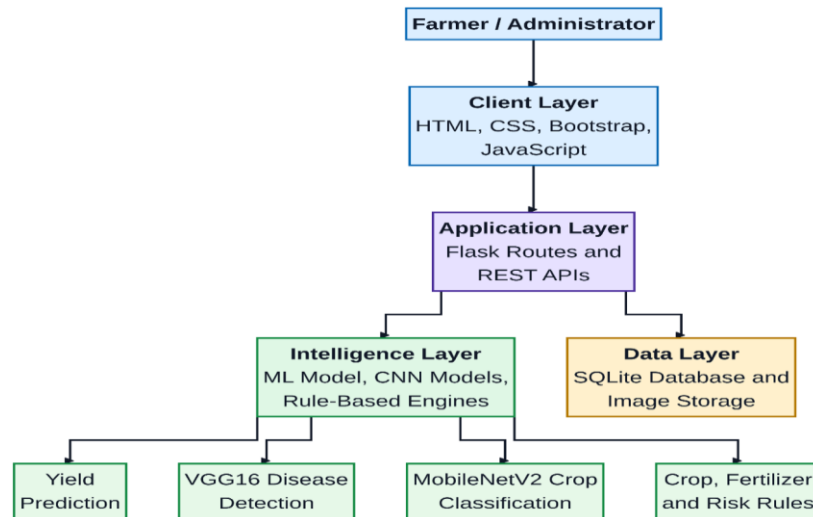


Figure 1 - System Architecture of AgriPredict

The client layer provides the interface, developed using HyperText Markup Language (HTML), Cascading Style Sheets (CSS), Bootstrap, and JavaScript. The Flask application receives requests, validates inputs, manages authentication, and directs requests to the appropriate service. The intelligence layer contains the crop-yield model, the VGG16 plant-disease model, the MobileNetV2 crop-classification model, and rule-based recommendation functions. SQLite stores user information, prediction history, forum posts, and marketplace listings.

3.2. Data Processing and Prediction Modules

The system processes numerical inputs and uploaded images based on the selected service. Crop-yield prediction uses crop type, rainfall, pesticide usage, and average temperature, with an optional NPK-based adjustment. Disease-risk assessment considers crop and environmental conditions, while crop and fertilizer recommendations use relevant soil, field, and farming information.

For image-based services, uploaded images are validated and preprocessed before analysis. VGG16 detects supported plant diseases, while MobileNetV2 classifies crop images. Results are displayed to users and can be saved in the prediction history of authenticated users.

Table 2 - Main AgriPredict modules and inputs

Module	Method or model	Main inputs or function
Crop-yield prediction	Machine-learning regression model	Crop type, rainfall, pesticide usage, temperature; optional NPK adjustment
Plant-disease detection	VGG16	Uploaded leaf image; predicts a supported disease or healthy-leaf class
Crop classification	MobileNetV2	Uploaded crop image; predicts a supported crop class
Disease-risk assessment	Predefined rules	Crop and weather-related conditions
Fertilizer recommendation	Predefined rules	Crop, soil, field size, growth stage, and nutrient information
Crop recommendation	Suitability rules	Soil, climate, water availability, farming experience, and market preference

The outputs depend on the information supplied, the classes represented in the trained image models, and the rules implemented in the application. They are intended as preliminary guidance, not as a replacement for expert advice or laboratory soil testing.

4. Data Flow and Implementation

4.1. Data Flow

AgriPredict accepts crop, soil, climate, farming, and image data through its web interface. The Flask application validates each request and directs it to the appropriate module for prediction or recommendation.

Numerical inputs are processed by the crop-yield model or rule-based modules. Yield prediction uses crop type, rainfall, pesticide usage, and average temperature, while disease-risk assessment considers crop and environmental conditions. Crop and fertilizer recommendations use relevant agricultural information.

For image-based services, uploaded images are resized and normalized before analysis. VGG16 detects plant diseases, while MobileNetV2 classifies crops. Results are displayed to users and stored in the SQLite database as prediction history for authenticated users. The database also manages user information, forum posts, and marketplace listings.

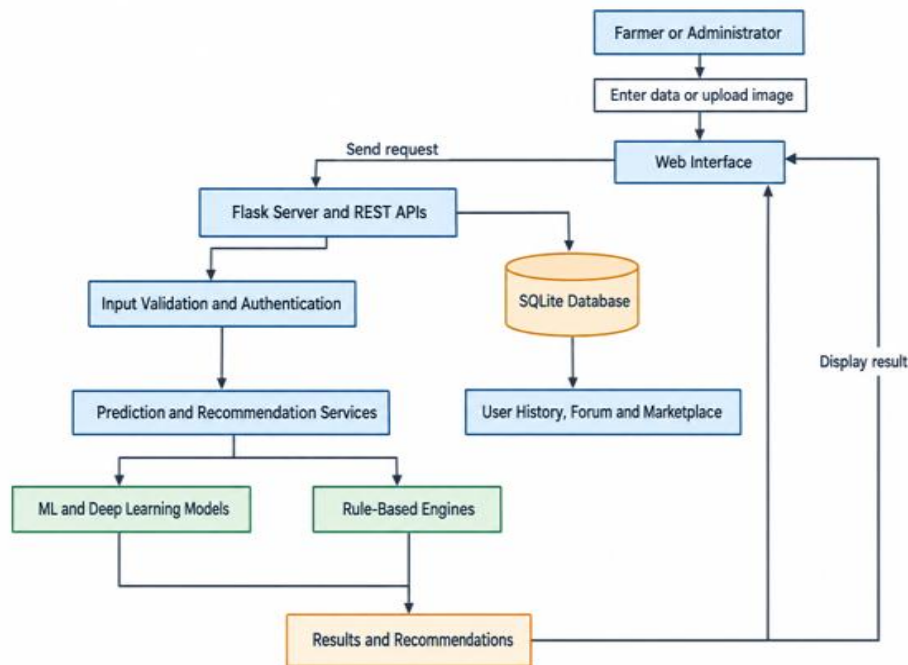


Figure 2 - Data Flow Diagram of AgriPredict

4.2. Software and Application Implementation

AgriPredict is implemented using Python and Flask for request handling, authentication, and input validation. The user interface uses HTML, CSS, Bootstrap, and JavaScript. TensorFlow/Keras and scikit-learn support model integration, while OpenCV, Pillow, and NumPy handle image preprocessing. SQLite stores application data.

The system includes modules for authentication, yield prediction, disease detection, crop classification, disease-risk assessment, crop and fertilizer recommendations, dashboards, farmer-connect features, and administration. Its modular design simplifies testing and maintenance.

4.3. Deep-Learning Model Integration

AgriPredict uses VGG16 for plant-disease detection and MobileNetV2 for crop classification. A legacy-compatible loading method is used for VGG16 to address Keras version compatibility, while MobileNetV2 uses the standard TensorFlow/Keras method. Both models use appropriate preprocessing, and their loading processes were tested separately without retraining. Predictions are limited to the classes represented in their respective training datasets. The predicted class is displayed through the web interface for user reference. This integration enables image-based analysis within the unified agricultural advisory system.

4.4. NPK-Based Yield Adjustment

The report describes an optional nitrogen, phosphorus, and potassium (NPK)-based adjustment to the initial crop-yield estimate. When NPK values are provided, the application checks them against approximate requirements for the selected crop and classifies the nutrient condition as low, moderate, or adequate. An adjustment factor is then applied to the initial prediction, and the adjusted estimate is displayed.

4.5. Validation and Data Management

The application validates submitted values before processing and checks whether uploaded images are supported and readable. The report describes exception handling for invalid inputs, missing model files, and database errors. These checks allow the application to handle common failures and display error messages. SQLite stores user details and prediction history, as well as forum posts and marketplace information. The report describes this database choice as suitable for an academic-scale application and identifies a production database, caching, model optimization, and cloud deployment as possible future enhancements.

5. Testing and Results

Testing was carried out to check whether AgriPredict's main functions operated as intended and whether the application's components worked together. The report describes unit testing of individual functions and integration testing of workflows involving the user interface, Flask routes, prediction and recommendation modules, and SQLite database. Tests covered both valid and invalid inputs.

5.1. Unit Testing

Unit testing was used to check individual application functions. The reported test cases included login, crop-yield prediction, image upload, plant-disease detection, fertilizer recommendation, and crop classification. For each function, the expected outcome was checked—for example, that valid inputs produced a result and that a supported image could be submitted for analysis. The report states that the tested functions produced their expected results.

5.2. Integration Testing

Integration testing checked whether related components exchanged information correctly when used as a workflow. Tests included logging in and opening the dashboard, submitting yield-prediction inputs, uploading images for VGG16 disease detection and MobileNetV2 crop classification, submitting soil information for fertilizer advice, assessing disease risk, requesting crop recommendations, and saving results to user history.

Table 3 - Integration test cases reported for AgriPredict

Test ID	Modules or workflow	Expected result reported
IT01	Login and dashboard	Dashboard opens
IT02	Yield form and prediction model	Yield is displayed
IT03	Image upload and VGG16 model	Disease result is displayed
IT04	Image upload and MobileNetV2 model	Crop class is displayed
IT05	Soil form and fertilizer module	Fertilizer advice is displayed
IT06	Prediction module and database	Result is saved
IT07	Weather input and disease-risk module	Risk level is displayed
IT08	Crop-recommendation module	Suitable crops are displayed

The report states that the integration tests confirmed that the modules exchanged data correctly and produced the expected outputs for the tested inputs. It also reports that prediction results were stored successfully in user history.

5.3. Input Validation and Error Handling

The report describes testing invalid values, empty fields, incorrect login details, and unsupported or unreadable image files. The application handled these cases by rejecting the input or displaying an appropriate error message. These checks covered common interaction and input-handling cases; they do not establish that every possible invalid input or failure condition was tested.

5.4. Performance Observations

Performance was assessed qualitatively by observing system responses. The report describes rule-based functions, including disease-risk assessment and recommendations, as responding quickly. Image-based services took longer because they require image preprocessing and deep-learning inference. The report characterizes the application's performance as acceptable for academic demonstration and small-scale use.

The report does not provide measured response times, dataset sizes, or quantitative model-evaluation results. Therefore, these findings should be presented as functional test outcomes and qualitative observations, not as measured predictive accuracy or evidence of performance at scale. Further evaluation would require documented test data and suitable metrics for each model.

Overall, the reported tests indicate that the main application workflows operated as intended for the inputs tested. The report identifies larger datasets, real-world agricultural inputs, and further model evaluation as areas for future work. Prediction quality remains dependent on input accuracy, image quality, the classes represented in the training data, and the rules incorporated into the application. The observed processing time can vary with the system resources available, model size, image resolution, and database operations. Image quality also affects the reliability of image-based predictions, while incomplete or inaccurate agricultural inputs may affect recommendations. The report therefore treats the system as an academic prototype for small-scale use. Broader evaluation would require testing with more users, diverse field images, and agricultural data from real operating conditions. Future evaluation

should use larger datasets and real-world agricultural inputs. Recording model metrics and response times would provide a more quantitative assessment of system performance.

6. Conclusion

AgriPredict integrates crop-yield prediction, crop classification, plant-disease detection, disease-risk assessment, and crop and fertilizer recommendations in one web application. It also provides weather information, user authentication, prediction history, a discussion forum, and a marketplace. The project demonstrates the integration of machine-learning, deep-learning, and rule-based services for preliminary agricultural decision support.

The report describes unit and integration testing of the system's main functions. The tested modules produced expected outputs for the inputs used, and the application handled several invalid inputs with suitable messages. However, the report does not provide numerical model-accuracy metrics or measured response times. The reported results therefore support functional operation, not quantified predictive performance.

6.1. Limitations and Future Work

The system's outputs depend on input accuracy, image quality, the classes represented in the trained models, and the rules used by the recommendation modules. The image services support only their trained classes. Fertilizer and crop recommendations are general guidance and should be checked against local conditions, soil-test results, and expert advice. AgriPredict is an academic prototype intended for small-scale use.

Future work may evaluate the models using larger and more diverse datasets, report quantitative metrics, expand the supported classes, and test the application using real-world agricultural data. Other extensions identified in the report include IoT-based soil sensors, regional-language support, mobile access, cloud deployment, and a production database. Broader testing would help establish how well the system works across different regions, crops, and farming conditions.

ACKNOWLEDGEMENTS

The authors sincerely thank Prof. Ullas H. S., Assistant Professor, Department of Computer Science and Engineering, Kalpataru Institute of Technology, for his guidance, valuable suggestions, and support throughout this project.

REFERENCES

- [1] Bala N, M., Nayak, T. K., Palla, J. S. S. P., Nath, S. G., & Jeswanth, S. (2025). Plant leaf disease detection using MobileNetV2. *ITM Web of Conferences*, 79, Article 01021.
- [2] Mahale, Y., Khan, N., Kulkarni, K., Wagle, S. A., Pareek, P., Kotecha, K., Choudhury, T., & Sharma, A. (2024). Crop recommendation and forecasting system for Maharashtra using machine learning with LSTM: A novel expectation-maximization technique. *Discover Sustainability*, 5, Article 134.
- [3] Manideep, A. P. S., & Kharb, S. (2022). A comparative analysis of machine learning prediction techniques for crop yield prediction in India. *Turkish Journal of Computer and Mathematics Education*, 13(2), pp. 120–133.
- [4] Paymode, A. S., & Malode, V. B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artificial Intelligence in Agriculture*, 6, pp. 23–33.
- [5] Nishant, P. S., Venkat, P. S., Avinash, B. L., & Jabber, B. (2020). Crop yield prediction based on Indian agriculture using machine learning. In *2020 International Conference for Emerging Technology (INCET)*.
- [6] Nevavuori, P., Narra, N., & Lipping, T. (2019). Crop yield prediction with deep convolutional neural networks. *Computers and Electronics in Agriculture*, 163, Article 104859.
- [7] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, pp. 311–318.
- [8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520.
- [9] Fuentes, J., Yoon, S., Kim, S., & Park, D. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), Article 2022.
- [10] Abadi, M., et al. (2016). TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*, pp. 265–283.
- [11] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- [12] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, Article 1419.
- [13] Sladojević, M., Arsenović, M., Anderla, A., Culibrk, D., & Stefanović, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, Article 3289801.
- [14] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- [15] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.