IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151129

Precise Crop Farming Using Machine Learning

Varad Sane[1], Varad Kulkarni[2], Shrikant Somani[3], Prashant Yanpallewar[4]

Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India[1-4]

Abstract: Agricultural production is influenced by spatial and temporal variations in crop health, soil moisture, temperature, humidity, and weather conditions. Conventional field inspection and fixed-schedule irrigation may not adequately represent these variations when decisions are required for individual crop zones. This paper presents the design and software implementation of an integrated precision crop-farming decision-support framework combining plant-image classification, weather information, configurable decision rules, and a software-based digital twin. The system employs MobileNetV3-Large as its plant-disease classification backbone. A React and Vite frontend communicates with a Java Spring Boot REST API, which coordinates a Python-based PyTorch inference service, PostgreSQL persistence, and the Open-Meteo weather service. The decision engine combines available crop-zone state, classification output, confidence score, and forecast context to generate one of four recommendations: Irrigate, Treat, Monitor, or Skip. A simulation component records virtual state transitions and maintains a history of decisions and actions. The proposed model-training configuration uses ImageNet-pretrained weights, two-phase fine-tuning, and PlantVillage as the baseline dataset, with PlantDoc identified for external robustness evaluation. The current work establishes a software workflow for integrating crop-health evidence and environmental context. Quantitative model evaluation, field validation, physical actuation, and measured resource-saving experiments have not yet been completed. The framework therefore provides an implementation baseline and a reproducible evaluation pathway for future precision-agriculture research.

Keywords: Precision agriculture, plant disease classification, MobileNetV3-Large, weather-aware irrigation, decision-support system, agricultural digital twin.

I. INTRODUCTION

A. Background and Motivation

Agriculture is a complex production system in which crop growth and productivity depend on interactions among biological, environmental, and management-related factors. Soil properties, moisture availability, temperature, humidity, sunlight, crop variety, disease pressure, and rainfall may vary considerably across a single field. Management practices that apply the same treatment or irrigation schedule to every part of a field may therefore fail to address local requirements.

Conventional agricultural monitoring frequently depends on manual inspection, periodic observation, and predefined irrigation schedules. These practices remain important, but can be limited when crop conditions change rapidly or when different field regions require different interventions. Machine learning, computer vision, Internet of Things technologies, and environmental data services provide complementary sources of information for precision agriculture. Machine-learning methods can analyse crop images, while sensors and weather services can provide environmental observations. Integrating these sources within a decision-support application can help organise information and make recommendations more traceable.

Machine learning has been applied to crop monitoring, disease recognition, yield estimation, and resource management. Condran et al. surveyed machine-learning applications in precision agriculture [6], while Saranya et al. examined the complementary roles of deep learning and Internet of Things technologies [7]. These developments motivate integrated systems that combine image-based analysis with environmental context rather than treating each prediction as an isolated output.

B. Problem Definition and Objectives

Plant-image classification alone does not provide sufficient information to determine every agricultural action. A model may identify a visual pattern associated with a disease, but an appropriate response may also depend on crop identity, disease confidence, the current condition of the crop zone, environmental factors, and expert guidance. Similarly, irrigation decisions should consider available moisture information and expected rainfall rather than relying exclusively on a fixed schedule.

This study addresses the software-system problem of integrating crop-image analysis, crop-zone information, weather context, rule-based decision-making, and state-history management in a unified application. The workflow allows a user to select a virtual crop zone, submit a plant image, receive a predicted class and confidence score, retrieve weather context, obtain a recommendation, and optionally simulate the corresponding action.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151129

The objectives are to provide crop-zone management, integrate a MobileNetV3-Large inference service, retrieve weather information, generate explainable Irrigate, Treat, Monitor, or Skip recommendations, and persist virtual state transitions. The system is intended to support agricultural decision-making and does not replace professional diagnosis or independently authorise pesticide application or physical irrigation.

C. Contributions

- A modular architecture connecting a web interface, application backend, machine-learning inference service, weather provider, and relational database.
- A context-aware rule-based workflow combining image-classification evidence, confidence, crop-zone information, and weather context.
- A configurable confidence gate that directs uncertain predictions toward monitoring or human review.
- A traceable software digital-twin representation that stores virtual state transitions and links simulated actions to decisions.
- A documented training and evaluation protocol for subsequent model and system validation.

II. RELATED WORK

A. Precision Agriculture and Decision Support

Precision agriculture uses information about field variability to support targeted crop management. Relevant information may include soil properties, crop condition, environmental measurements, historical observations, and weather forecasts. Machine-learning techniques are increasingly used to process agricultural data and identify patterns that may support management decisions [6], [7].

A distinction exists between prediction and decision-making. A predictive model estimates a condition, whereas a decision-support system combines that estimate with operational constraints and management objectives. Identifying a possible disease from a plant image does not independently establish the appropriate treatment. The result must be interpreted in the context of crop identity, prediction confidence, disease severity, and agronomic recommendations. The proposed framework therefore treats image classification as one evidence source within a larger decision process.

B. Computer Vision for Plant Disease Detection

Computer vision has been widely investigated for identifying plant diseases from leaf and crop images. Convolutional neural networks learn visual representations through convolutional operations, nonlinear activation functions, and feature aggregation. Mohanty et al. demonstrated deep learning for image-based plant-disease detection using PlantVillage [1]. Their results demonstrated the potential of CNN-based classification under controlled image conditions, but controlled-image performance should not be interpreted as equivalent to field performance.

Singh et al. introduced PlantDoc as a dataset for visual plant-disease detection using more varied images [2]. Field-style photographs may contain complex backgrounds, changing illumination, occlusion, and variations in image quality. Jelali reviewed deep-learning-based tomato disease and pest detection using real-field datasets and discussed limitations in field-oriented evaluation [3]. These studies motivate separate internal and external evaluation.

Plant disease recognition can be formulated as image classification, object detection, or segmentation. Classification assigns a label to an entire image; object detection additionally identifies object locations; segmentation identifies affected image regions. This project uses whole-image classification because its primary requirement is to obtain a crop-health category for the selected zone.

C. Lightweight Networks and Dataset Strategy

Howard et al. introduced MobileNets as efficient convolutional neural networks for mobile vision applications [8]. MobileNet architectures use depthwise separable convolutions to reduce computational requirements. MobileNetV3, introduced by Howard et al. [19], incorporates hardware-aware design considerations. EfficientNet, proposed by Tan and Le [9], provides an alternative approach to scaling network dimensions. MobileNetV3-Large is selected as the baseline classifier in this project.

PlantVillage is identified as the baseline training dataset [1], while PlantDoc is intended for external robustness evaluation [2]. Dataset version, class distribution, image counts, licence, and preprocessing should be documented. Duplicate or related images must not leak across training and testing partitions. PlantDoc classes must be mapped explicitly to the model's output classes before external evaluation.

D. IoT Sensing and Weather-Aware Irrigation

Internet of Things technologies enable collection and communication of environmental measurements through networked sensors. Agricultural sensing systems may include soil-moisture probes, temperature and humidity sensors, rainfall measurements, and water-flow sensors. Shaikh et al. reviewed IoT-enabled sensor technologies for smart agriculture [10],

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151129

while Soussi et al. discussed smart sensors and data quality for precision agriculture [11]. Raw soil-moisture readings require calibration because soil composition, salinity, temperature, and sensor placement influence interpretation.
Irrigation scheduling approaches include fixed-interval methods, moisture thresholds, evapotranspiration-based calculations, predictive models, and hybrid strategies. Togneri et al. surveyed data-driven water-need estimation for IoT-based smart irrigation [4], and Ivanova reviewed soil-moisture forecasting using sensor, weather, and irrigation observations [5]. FAO-56 describes reference evapotranspiration and crop coefficients for estimating crop water requirements [12]. Weather forecasts can provide additional context, but forecast uncertainty and local microclimate must be considered.

E. Agricultural Digital Twins and Research Gap

Agricultural digital twins have been explored for crop monitoring, farm operations, and resource management [13]–[15]. A digital representation should define its entities, state variables, update mechanisms, data sources, and simulation assumptions. A dashboard alone is not necessarily a validated digital twin. The present system maintains a software representation of virtual crop zones and records state transitions; it is not continuously synchronised with a physical field and is not a validated crop-growth model.
The reviewed literature establishes the relevance of computer vision, environmental sensing, irrigation scheduling, and digital representations. The system developed here addresses an integration-oriented gap by connecting image-based crop-health analysis, weather information, zone-level decision rules, and persistent virtual state transitions within one modular software workflow. The contribution is system integration and traceability, not a new classification algorithm or a validated agronomic model.

III. PROPOSED SYSTEM ARCHITECTURE AND METHODOLOGY

A. System Overview

The proposed system is organised into four logical layers: (i) presentation, (ii) application and decision, (iii) intelligence and external services, and (iv) data persistence and simulation. The layers communicate through defined interfaces, separating user interaction from application orchestration, model inference, weather retrieval, and data storage.

B. Presentation Layer

The presentation layer is implemented using React and Vite. It provides the user-facing interface for selecting virtual crop zones, inspecting zone details, uploading plant images, viewing classification results and confidence scores, reviewing weather information, examining recommendations, initiating virtual simulations, and viewing historical records. The interface should distinguish externally retrieved, user-entered, and simulated values so that virtual values are not mistaken for field measurements.

C. Application and Decision Layer

The application layer is implemented using Java Spring Boot and exposes REST endpoints to the frontend. Its responsibilities include request validation, workflow orchestration, communication with the inference service, weather retrieval, decision evaluation, database operations, and simulation coordination. The backend validates the crop-zone identifier and image request before invoking the machine-learning service. It subsequently combines the model output with zone and weather information.

D. Intelligence and External-Service Layer

The intelligence layer contains a Python-based PyTorch inference service using MobileNetV3-Large. It receives a validated plant image, performs preprocessing, executes model inference, and returns a predicted class, confidence score, and model version. Weather integration uses the Open-Meteo Forecast API [18]. The backend requests forecast information using the location associated with the selected crop zone. If the provider is unavailable, the application records an unavailable state rather than substituting fabricated values.

E. Data Persistence and Entity Relationships

PostgreSQL stores five principal entities: Zone, Analysis, Weather, Decision, and Simulation. A Zone can have multiple Analysis, Weather, and Decision records. Each Analysis belongs to one zone. A Decision may reference one Analysis and one Weather record; these references are nullable when evidence is unavailable or not applicable. A Simulation references the Decision that initiated the simulated action, and its zone association is derived through that Decision.

 This work is licensed under a Creative Commons Attribution 4.0 International License

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151129

TABLE I. RELATIONAL DATA ENTITIES

Entity	Important Attributes	Purpose
Zone	zone_id, name, crop_type, moisture_value, moisture_unit, temperature, humidity, health_status, state_source, updated_at	Maintains virtual crop-zone state, units, and provenance.
Analysis	analysis_id, zone_id, image_ref, predicted_label, confidence, model_version, timestamp	Stores image-analysis output and model metadata.
Weather	weather_id, zone_id, forecast_data, source, captured_at	Preserves forecast snapshots and source information.
Decision	decision_id, zone_id, analysis_id, weather_id, action, rationale, timestamp	Stores recommendation and evidence references.
Simulation	simulation_id, decision_id, before_state, after_state, estimated_saving	Records virtual action and before/after state.

Figure 1. Overall architecture showing the presentation, application and decision, intelligence and external-service, and persistence and simulation layers.
Figure 2. Conceptual data-flow diagram showing zone selection, image analysis, weather retrieval, decision evaluation, simulation, and history persistence.
Figure 3. Entity-relationship model containing Zone, Analysis, Weather, Decision, and Simulation entities.

F. MobileNetV3-Large Training Configuration

The plant-disease classifier uses MobileNetV3-Large with ImageNet-pretrained weights [19]. The final classification layer is replaced with a task-specific output layer corresponding to the disease and healthy categories represented in the training dataset. The documented training configuration is summarised in Table I. These settings describe the intended training procedure and must be confirmed against actual training logs before being reported as executed experiments.

TABLE II. DOCUMENTED MOBILE-NETV3-LARGE TRAINING CONFIGURATION

Parameter	Specified Configuration
Backbone and initial weights	MobileNetV3-Large; ImageNet-pretrained
Input	224 × 224 RGB; ImageNet mean and standard-deviation normalisation
Phase 1	Backbone frozen; classifier head trained for 3 epochs at learning rate 0.001
Phase 2	All layers fine-tuned for 10 epochs at learning rate 0.0001
Optimiser and schedule	AdamW; weight decay 0.0001; cosine learning-rate schedule
Loss	Class-weighted cross-entropy; label smoothing 0.05
Batch size and seed	32; fixed seed 42
Data partition	70% training, 15% validation, 15% testing
Model selection and export	Best validation-accuracy checkpoint; ONNX export

G. Training Procedure and Preprocessing

Training is specified in two phases. In Phase 1, the pretrained backbone is frozen and the new classification head is trained for three epochs using a learning rate of 0.001. In Phase 2, the backbone is unfrozen and the complete network is fine-tuned for ten epochs using a learning rate of 0.0001. AdamW is specified with weight decay 0.0001 and a cosine learning-rate schedule. Class-weighted cross-entropy with label smoothing of 0.05 is used to address class imbalance.
The inference pipeline converts the image to RGB, resizes it, applies a centre crop to 224 × 224, and normalises pixel values using ImageNet statistics. Training-only augmentation includes random cropping, flipping where appropriate, rotation, colour jitter, blur, and random erasing. Validation and test images must not undergo random training augmentation.

H. Confidence Gate and Decision Logic

The inference service returns a predicted class, confidence score, model version, timestamp, and highest-probability classes. The softmax confidence is a model score and should not be interpreted as a calibrated probability unless a

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151129

calibration procedure is performed. The documented default confidence threshold is 0.70. Predictions below the configured threshold are treated as uncertain and result in Monitor.

The decision engine generates four outcomes. Irrigate is considered when the configured zone-moisture condition indicates a need and forecast rainfall does not justify deferral. Treat is considered when a disease-related category is predicted with sufficient confidence and configured conditions are satisfied. Monitor is returned when evidence is incomplete, confidence is low, or the information does not support a reliable intervention. Skip is returned when the available evidence indicates that no immediate simulated action is required. Thresholds are configurable and require crop-specific agronomic review.

The decision process can be represented as $D(t) = f(Z(t), P(t), C(t), W(t), R)$, where $Z(t)$ is zone state, $P(t)$ is the predicted category, $C(t)$ is confidence, $W(t)$ is weather context, and R represents configured rules. This is a conceptual representation of the rule-based process, not a learned decision model.

I. Software Digital-Twin Simulation

The simulation component maintains a virtual representation of each crop zone. Its state may include crop identity, moisture, temperature, humidity, health status, and recent actions. A simulated action updates the virtual state according to configured transition rules. The general transition is $X(t+1) = G(X(t), A(t), C(t))$, where $X(t)$ is virtual state, $A(t)$ is the simulated action, $C(t)$ is available context, and G is the configured transition function.

A simplified virtual moisture update is $M(t+1) = \max(0, \min(Mmax, M(t) + \Delta Maction - \Delta Mloss))$. The coefficients are simulation assumptions and require calibration against observations before the simulation can be treated as representative of physical crop-zone behaviour. Simulation records store before-and-after states and link each simulated action to its originating decision.

A theoretical water-saving calculation may be expressed as $S(\%) = ((Vbaseline - Vscenario) / Vbaseline) \times 100$. This calculation is meaningful only when both volumes use the same time interval, zone boundary, and measurement basis. Without flow-meter measurements, such values are estimates and must be labelled accordingly. No measured water-saving result is claimed.

IV. RESULTS AND DISCUSSION

A. Implementation Scope

The project report documents a software implementation integrating a React/Vite frontend, Java Spring Boot backend, PostgreSQL database, Python-based MobileNetV3-Large inference component, and Open-Meteo weather integration. The application workflow connects crop-zone selection, image submission, classification, weather retrieval, decision generation, simulation, and history persistence. These are implementation capabilities and do not establish measured improvements in disease-detection performance, irrigation efficiency, crop productivity, or resource consumption.

B. Machine-Learning Evaluation Status

The MobileNetV3-Large training configuration specifies a 70:15:15 dataset partition, two-phase fine-tuning, AdamW optimisation, and a held-out evaluation procedure. However, the model has not yet been trained and evaluated. Consequently, no measured test accuracy, precision, recall, macro-F1, confusion matrix, or inference latency is available. Such values are not reported in this manuscript.

The following metrics are specified for the subsequent experiment. Accuracy is the number of correct predictions divided by the total number of test samples. For class i, precision is $TP_i/(TP_i+FP_i)$, recall is $TP_i/(TP_i+FN_i)$, and $F1_i$ is $2(Precision_i \times Recall_i)/(Precision_i+Recall_i)$. Macro-F1 is the arithmetic mean of the per-class F1 scores. Macro-F1 and per-class recall are important when class frequencies are unequal.

C. Confusion-Matrix and External Evaluation

A confusion matrix should be generated using the held-out test partition, with actual classes represented by rows and predicted classes by columns. It can reveal disease categories that are frequently confused. Per-class recall is particularly important because low recall may indicate that affected plants are classified as healthy or assigned to another disease category.

PlantDoc should be used for a separate external robustness evaluation. The experiment should document the number of evaluated images, class mapping, excluded categories, and differences in label definitions. Performance differences between the PlantVillage test set and PlantDoc can provide information about sensitivity to lighting, background, occlusion, and image quality. External evaluation should remain independent of model selection.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151129

D. Decision-Support and Simulation Discussion

The decision engine is designed to combine information that would otherwise be considered independently. The confidence gate directs uncertain predictions toward monitoring rather than automatically triggering treatment. Weather information provides additional context for irrigation-related decisions, but forecasts do not directly measure soil moisture in an individual crop zone. The reliability of an irrigation recommendation therefore depends on the quality and availability of zone-state information and the configured rules.

The software digital twin records virtual crop-zone changes and links before-and-after state information to the originating decision. Its consistency should be assessed by checking that state values remain within configured limits, each simulation references a valid decision, and repeated operations produce expected records. The simulation does not establish that an equivalent physical action would produce the same moisture or crop-health response.

E. Limitations

Quantitative model evaluation has not been completed. PlantVillage-based training alone may not provide adequate generalisation to field images. Weather forecasts do not provide direct soil-moisture measurements. The decision engine uses configurable rules whose thresholds require crop-specific agronomic review. The current implementation does not include physical ESP32 sensing, pump control, automated spraying, or continuous synchronisation with a field. Water-saving and crop-yield improvements have not been measured. These limitations restrict the conclusions to the software implementation and proposed evaluation methodology.

V. CONCLUSION AND FUTURE WORK

A. Conclusion

This paper presents the design and software implementation of an integrated precision crop-farming decision-support framework connecting plant-image classification, weather information, rule-based recommendations, and a software digital-twin simulation. The system uses a React/Vite frontend, Java Spring Boot REST API, PostgreSQL database, Python/PyTorch inference service based on MobileNetV3-Large, and Open-Meteo weather integration.

Its decision engine produces Irrigate, Treat, Monitor, and Skip outcomes. A confidence-based mechanism directs uncertain predictions toward monitoring or human review. The simulation component records virtual crop-zone state transitions and maintains a history of decisions and simulated actions. The principal contribution is the integration of these components into a modular and traceable software workflow. The current work establishes an implementation baseline and an evaluation protocol; it does not establish quantitative classification performance, field-level irrigation effectiveness, measured water savings, or crop-yield improvements.

B. Future Work

- Train the MobileNetV3-Large model using the documented configuration and report accuracy, precision, recall, macro-F1, per-class metrics, and a confusion matrix.
- Evaluate robustness using PlantDoc and locally collected images with appropriate permissions and reliable annotations.
- Compare MobileNetV3-Large with alternative architectures under consistent dataset partitions, reporting model size and inference latency.
- Integrate an ESP32-based sensing module for soil-moisture, temperature, and humidity observations, with calibration for the target soil and installation conditions.
- Develop and separately test a physical irrigation interface with safety checks and explicit authorisation.
- Calibrate digital-twin transition parameters using observed measurements and compare simulated moisture changes with sensor readings.
- Conduct controlled field-scale validation using defined crop zones, documented observations, and a clearly specified baseline.
- Measure water consumption using suitable flow meters and review decision thresholds with qualified agricultural specialists.

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, Art. no. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[2]. D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, “PlantDoc: A dataset for visual plant disease detection,” in Proc. 7th ACM IKDD CoDS and 25th COMAD, 2020, pp. 249–253, doi: 10.1145/3371158.3371196.

 This work is licensed under a Creative Commons Attribution 4.0 International License



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151129

[3]. M. Jelali, "Deep learning networks-based tomato disease and pest detection: A first review of research studies using real field datasets," Frontiers in Plant Science, vol. 15, Art. no. 1493322, 2024, doi: 10.3389/fpls.2024.1493322.
[4]. R. Togneri, R. Prati, H. Nagano, and C. Kamienski, "Data-driven water need estimation for IoT-based smart irrigation: A survey," Expert Systems with Applications, vol. 225, Art. no. 120194, 2023, doi: 10.1016/j.eswa.2023.120194.
[5]. I. Ivanova, "Soil moisture forecasting from sensors-based soil moisture, weather and irrigation observations: A systematic review," Smart Agricultural Technology, vol. 10, Art. no. 100692, 2025, doi: 10.1016/j.atech.2024.100692.
[6]. S. Condran, M. Bewong, M. Z. Islam, L. Maphosa, and L. Zheng, "Machine learning in precision agriculture: A survey on trends, applications and evaluations over two decades," IEEE Access, vol. 10, pp. 73786–73803, 2022, doi: 10.1109/ACCESS.2022.3188649.
[7]. T. Saranya, C. Deisy, S. Sridevi, and K. S. M. Anbananthen, "A comparative study of deep learning and Internet of Things for precision agriculture," Engineering Applications of Artificial Intelligence, vol. 122, Art. no. 106034, 2023, doi: 10.1016/j.engappai.2023.106034.
[8]. A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017, doi: 10.48550/arXiv.1704.04861.
[9]. M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. 36th International Conference on Machine Learning, PMLR, vol. 97, pp. 6105–6114, 2019.
[10]. F. K. Shaikh, S. Karim, S. Zeadally, and J. Nebhen, "Recent trends in Internet-of-Things-enabled sensor technologies for smart agriculture," IEEE Internet of Things Journal, vol. 9, no. 23, pp. 23583–23598, 2022, doi: 10.1109/JIOT.2022.3210154.
[11]. A. Soussi, E. Zero, R. Sacile, D. Trinchero, and M. Fossa, "Smart sensors and smart data for precision agriculture: A review," Sensors, vol. 24, no. 8, Art. no. 2647, 2024, doi: 10.3390/s24082647.
[12]. R. G. Allen, L. S. Pereira, D. Raes, and M. Smith, Crop Evapotranspiration: Guidelines for Computing Crop Water Requirements. Rome: FAO, Irrigation and Drainage Paper 56, 1998.
[13]. L. Wang, "Digital twins in agriculture: A review of recent progress and open issues," Electronics, vol. 13, no. 11, Art. no. 2209, 2024, doi: 10.3390/electronics13112209.
[14]. A. C. Tagarakis et al., "Digital twins in agriculture and forestry: A review," Sensors, vol. 24, no. 10, Art. no. 3117, 2024, doi: 10.3390/s24103117.
[15]. M. Escribà-Gelonch et al., "Digital twins in agriculture: Orchestration and applications," Journal of Agricultural and Food Chemistry, vol. 72, no. 19, pp. 10737–10752, 2024, doi: 10.1021/acs.jafc.4c01934.
[16]. B. Nsoh et al., "Internet of Things-based automated solutions utilizing machine learning for smart and real-time irrigation management: A review," Sensors, vol. 24, no. 23, Art. no. 7480, 2024.
[17]. M. N. Mowla et al., "Internet of Things and wireless sensor networks for smart agriculture applications: A survey," IEEE Access, vol. 11, pp. 145813–145852, 2023, doi: 10.1109/ACCESS.2023.3346299.
[18]. Open-Meteo, "Weather Forecast API." [Online]. Available: https://open-meteo.com/en/docs. Accessed: Nov. 1, 2026.
[19]. A. Howard et al., "Searching for MobileNetV3," in Proc. IEEE/CVF International Conference on Computer Vision, 2019, pp. 1314–1324.

 This work is licensed under a Creative Commons Attribution 4.0 International License