International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

An AI-Based Plant Disease Detection and Agri E-Commerce System using Flask and MongoDB

S.Sasikala,M.E., (Ph. D)., Gokularamanan J, Jothikumar R, Jana Krishnan D, Karthik T

Department of Computer Science and Engineering, R P Sarathy Institute of Technology, Affiliated to Anna University, Chennai, Salem, Tamil Nadu, India

Publication History: Received: 25.02.2026; Revised: 20.03.2026; Accepted: 25.03. 2026; Published: 28.03.2026.

ABSTRACT: Agriculture plays a vital role in the economic development of many countries, yet farmers often face significant challenges in identifying crop diseases and accessing quality agricultural products in a timely manner. This paper proposes AgroNest, an AI-Based Plant Disease Detection and Agri E-Commerce System that integrates modern deep learning technology with secure online commerce to support farmers in both crop health management and agricultural product purchasing. The platform provides a web-based interface where farmers can browse, search, and securely purchase agricultural inputs such as seeds, fertilizers, and farming tools. A deep learning-based plant disease detection module allows users to upload images of plant leaves for automated analysis using the YOLOv8 object detection model, which accurately identifies and classifies up to 28 distinct plant disease categories, returning annotated results with bounding boxes and confidence scores. The platform incorporates complete e-commerce functionality including a shopping cart, order management, and payment processing, alongside an automated SMS notification feature powered by the Twilio API that dispatches treatment reminders 48 hours after disease detection. The backend is developed using Flask (Python), MongoDB for flexible NoSQL data management, and Redis for caching and task queuing. Security is enforced through Werkzeug password hashing, JWT-based session management, and role-based access control. Experimental evaluation on 3,000+ labeled leaf images confirmed a mean Average Precision (mAP@0.5) exceeding 91%, while user testing with 30 farmers over three weeks demonstrated high usability, secure transaction handling, and effective notification delivery. The results demonstrate that AI can be practically and affordably integrated into agricultural platforms, contributing meaningfully to precision farming, food security, and sustainable rural development.

KEYWORDS: Plant Disease Detection, YOLOv8, Deep Learning, Agri E-Commerce, Flask, MongoDB, Precision Agriculture, SMS Notification, Twilio API, Object Detection, Convolutional Neural Networks

I. INTRODUCTION

Agriculture remains the backbone of the global economy, directly supporting more than 600 million smallholder farming households worldwide and providing livelihoods to approximately 1.3 billion people across developing nations [1]. Despite its critical importance, the agricultural sector continues to suffer from persistent structural inefficiencies that dramatically reduce crop productivity and farmer income. Among the most consequential of these inefficiencies are the delayed and inaccurate identification of plant diseases and the limited access farmers have to quality agricultural inputs at fair and transparent prices.

Plant diseases alone account for annual global crop losses estimated between 20% and 40% of total agricultural production, translating to hundreds of billions of dollars in economic damage each year [2]. Traditionally, disease diagnosis has depended on visual inspection by farmers themselves or consultation with local agricultural extension officers, a process that is inherently time-consuming, geographically constrained, and prone to human error. By the time a disease is correctly identified and treatment procured, the infection may have already spread to neighbouring crops, compounding the damage.

Simultaneously, the procurement of seeds, fertilizers, pesticides, and farming tools has remained tethered to physical markets beset by geographical limitations, inconsistent pricing driven by intermediary inefficiencies, and limited visibility into product availability and quality. While digital platforms for agriculture have begun to emerge, they are typically fragmented in scope, offering either disease detection tools or e-commerce functionality in isolation but rarely both within a coherent, unified system.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

This paper presents AgroNest, a web-based platform that bridges these critical gaps by integrating three tightly coupled subsystems: (i) an AI-powered plant disease detection engine built on YOLOv8, (ii) a full-featured agricultural e-commerce marketplace, and (iii) an automated SMS-based treatment reminder service via the Twilio API. AgroNest empowers farmers to diagnose crop diseases within seconds by uploading a photograph of an affected leaf, receive annotated results with disease classification and confidence scores, purchase the appropriate treatment products directly on the platform, and receive a follow-up SMS reminder 48 hours later to apply the treatment.

The system is developed with Flask (Python) on the backend, MongoDB as the primary data store, and a responsive HTML/CSS/JavaScript frontend rendered through Flask's Jinja2 template engine. Security is a first-class concern, implemented through hashed password storage, JWT authentication, and role-based access control distinguishing between farmer and administrator accounts. The design process followed a User-Centered Design (UCD) methodology, ensuring the interface remains accessible to farmers with minimal digital literacy and varying levels of smartphone experience.

The primary contributions of this work are: (i) an end-to-end integration of YOLOv8 object detection for real-time, multi-class plant disease recognition within a web application; (ii) a secure, scalable agri e-commerce module with product management, shopping cart, and order tracking; (iii) an automated asynchronous SMS notification pipeline using Celery and Twilio; (iv) a farmer-centric UCD approach validated through real-world deployment with 30 participants; and (v) a comprehensive security framework designed for first-time digital users in agricultural contexts.

II. LITERATURE REVIEW

A. AI-Based Disease Detection in Agriculture

The application of artificial intelligence to plant disease detection has evolved rapidly over the past decade. Landmark work by Mohanty et al. (2016) demonstrated that convolutional neural networks (CNNs) trained on the PlantVillage dataset could achieve classification accuracy exceeding 99% under controlled laboratory conditions [2]. This foundational study established the viability of image-based automated plant health monitoring and catalysed a wave of subsequent research. Ferentinos (2018) extended this work by evaluating multiple deep CNN architectures across 58,000 images representing 26 diseases and confirmed the strong generalisability of deep learning for this task [3].

However, early classification-focused work had a critical limitation: it assumed a single disease per image. In practice, multiple diseases may appear simultaneously on a single plant, and the disease regions may occupy only a small fraction of the image. To address this, researchers adapted object detection frameworks. Sladojevic et al. (2016) applied deep neural network classifiers to segment disease regions [4]. More recent work has leveraged single-stage detectors from the YOLO family for real-time plant disease detection, achieving strong performance under varied field conditions including differing lighting, occlusion, and background complexity [5]. YOLOv5 and YOLOv7 implementations have demonstrated mAP values between 78% and 88% on publicly available plant disease detection datasets. The present work employs YOLOv8, the current state-of-the-art iteration, which introduces an anchor-free detection head and improved feature pyramid architecture that together deliver superior performance, particularly for small lesion regions on large leaves [14]

B. Deep Learning Models for Plant Disease Identification

The literature documents a clear transition from classical machine learning to sophisticated deep learning architectures for plant disease identification. Traditional pipelines relied on hand-engineered features—colour histograms, texture descriptors, and morphological filters—fed into classifiers such as Support Vector Machines (SVM) or k-Nearest Neighbours (k-NN) [6]. While these approaches worked reasonably well under controlled conditions, they were fragile when applied to field images characterised by variable lighting, cluttered backgrounds, and overlapping lesions.

Deep CNNs transformed the landscape by learning hierarchical feature representations directly from raw pixel data. Too et al. (2019) conducted a rigorous comparative study of VGG16, Inception V4, ResNet50, and DenseNet201 for plant disease identification, finding that DenseNet201 achieved the highest accuracy (99.75%) while MobileNet offered the best trade-off between accuracy and computational efficiency, a critical consideration for deployment on resource-constrained devices [7]. Transfer learning studies have further demonstrated that fine-tuning ImageNet-pretrained models on domain-specific plant disease datasets substantially outperforms training from scratch, particularly when labelled agricultural data is scarce [8]. This finding directly motivated the transfer-learning-based training strategy employed for the YOLOv8 model in AgroNest, where ImageNet-pretrained backbone weights were fine-tuned on the domain-specific leaf disease dataset.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

C. E-Commerce in Agricultural Marketplaces

The integration of e-commerce into agricultural supply chains has attracted significant scholarly attention, particularly in developing economies where traditional market structures impose the greatest friction on smallholder farmers. Aker (2010) demonstrated through field experiments in Niger that mobile phone-based market information services reduced price dispersion across markets and improved farmer welfare [9]. Digital marketplace platforms have since emerged as mechanisms for connecting farmers directly with input suppliers and buyers, bypassing traditional intermediaries and their associated transaction costs.

The Government of India's e-NAM (National Agriculture Market) initiative, analysed by Mittal et al. (2018), demonstrated measurable improvements in price discovery and market reach for smallholder farmers through an online trading platform [10]. However, research consistently highlights ongoing challenges including inadequate logistics infrastructure, low digital literacy among older farming populations, and significant trust deficits arising from past negative experiences with digital fraud [20]. Security considerations are particularly salient: Zhao et al. (2020) demonstrated the application of cryptographic safeguards for protecting user data and transaction integrity in agricultural platforms [11]. These findings directly inform the security architecture of AgroNest, which prioritises hashed credential storage, JWT session management, and encrypted communication over HTTPS.

D. Automated Notification Systems for Agriculture

Timely communication is critically important in agricultural decision-making. Studies in agricultural informatics have consistently shown that automated notification systems—whether delivered via SMS, mobile apps, or IoT-connected devices—can significantly improve farmer compliance with recommended practices including irrigation scheduling, fertiliser application, and disease treatment timelines [12]. Krishna et al. (2018) evaluated SMS-based weather advisory services in rural India and reported substantial adoption even among farmers using basic feature phones, highlighting the appropriateness of SMS as a communication channel for the target demographic [13]. Research integrating IoT sensors with cloud platforms has further demonstrated that proactive alert systems with scheduled reminders substantially outperform passive information provision in changing farmer behaviour. The 48-hour post-detection SMS reminder design in AgroNest is directly informed by these findings.

E. Security and Data Protection

Web systems handling personal data—particularly those serving first-time digital users—require robust security design to mitigate threats including unauthorised access, credential theft, cross-site scripting (XSS), and NoSQL injection attacks. Han et al. (2020) outlined best practices for securing MongoDB deployments, including input validation, parameterised query patterns, and authentication enforcement at the database level [19]. Vieira et al. (2018) established comprehensive guidelines for secure session management in Flask-based web applications [18]. Campbell and Jackson (2021) established that bcrypt-based and PBKDF2-SHA256-based password hashing algorithms substantially reduce exposure to brute-force attacks and credential leakage compared to MD5 or SHA-1 hashing. These findings collectively shaped the multi-layer security architecture implemented in AgroNest.

III. METHODOLOGY

A. Research Approach

The development of AgroNest followed a four-phase sequential exploratory design combining literature review, prototype evaluation, iterative user feedback, and empirical system validation. Phase 1 (Exploratory Research) reviewed 28 research papers and conducted semi-structured interviews with 12 agricultural experts and contextual observations of 20 farmers to identify pain points in disease diagnosis and market access. Phase 2 (Requirements Gathering) employed focus groups with 10 farmers, a 200-respondent survey on digital adoption and disease incidence, and evaluation of existing platforms (e-NAM, FarmDrive). Phase 3 (Iterative Design and Testing) developed wireframes and high-fidelity prototypes, tested with 25 farmers using think-aloud protocols, and conducted technical accuracy tests for the YOLOv8 model across varying image conditions. Phase 4 (Evaluation and Validation) deployed the complete system with 30 farmers over three weeks, measuring detection accuracy, user satisfaction, e-commerce task completion rates, and SMS delivery confirmation.

B. Agile Development Methodology

The project employed an Agile-based development framework with two-week sprints enabling continuous integration, iterative testing, and adaptive improvement. Five development phases structured the overall project: (1) Foundation—Flask/MongoDB setup, JWT authentication, and role-based access control; (2) Core E-Commerce—product catalogue, paginated browsing, shopping cart, and order management; (3) AI Detection—YOLOv8 model training, integration,

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

and image preprocessing pipeline; (4) Notification System—Celery task scheduling, Twilio SMS integration, and notification logging; and (5) Refinement and Deployment—security audit, performance optimisation, Redis caching integration, and cloud hosting with HTTPS. Parallel development tracks for the AI module, e-commerce module, and SMS scheduling were maintained with synchronised integration checkpoints at each sprint boundary.

C. User-Centered Design Process

The UCD process followed five phases aligned with the ISO 9241-210 standard for human-centred design of interactive systems. The Discover phase mapped existing farmer workflows for crop disease management and product procurement through direct observation. The Define phase created six detailed farmer personas reflecting different crops, regions, and digital skill levels, and developed journey maps highlighting diagnostic and purchasing bottlenecks. The Ideate phase used co-design workshops with farmers and agricultural extension officers to generate candidate features, of which image-based disease detection, SMS treatment reminders, and simplified product search were consistently ranked highest. The Prototype phase produced low-fidelity wireframes followed by interactive HTML/CSS prototypes. The Test and Refine phase conducted iterative usability sessions with real farmers, capturing task success rates, error frequencies, and satisfaction ratings before finalising the production interface.

IV. SYSTEM ARCHITECTURE

A. High-Level Architecture

AgroNest adopts a modular, four-layer architecture cleanly separating the Presentation Layer, Application Layer, AI Processing Layer, and Data Layer. The Presentation Layer delivers a responsive web interface using HTML5, CSS3, JavaScript, and Flask's Jinja2 template engine. The Application Layer, implemented as Flask Blueprints, handles user authentication, product and order management, and external API coordination. The AI Processing Layer routes submitted leaf images through an OpenCV preprocessing pipeline to a YOLOv8 model and returns annotated results. The Data Layer uses MongoDB for persistent storage and Redis for caching and Celery task queuing. A Notification Service integrates the Twilio REST API to dispatch scheduled SMS reminders, and Flask-SocketIO provides real-time WebSocket push notifications for live inventory updates.

B. Frontend Architecture

The frontend adopts a component-based design in which each functional unit—login form, product grid, image upload panel, detection results display, shopping cart, and order history—is implemented as a self-contained Jinja2 template fragment. The disease detection component features a drag-and-drop image upload interface with real-time file preview, an animated progress indicator during model inference, and a side-by-side display of the original and annotated images alongside a textual summary of detected diseases and confidence percentages. Responsive CSS media queries ensure compatibility across desktops, tablets, and mobile browsers. Two-language support (English and Tamil) is provided via the Google Translate API embedded in the navigation bar, enabling accessibility for Tamil-speaking farmers in the primary deployment region of Tamil Nadu.

C. Backend Architecture

The Flask backend is organised as five microservice-aligned Blueprints: User Management (registration, login, hashed password storage, profile updates), Product Management (CRUD operations, inventory tracking, search), Order Processing (cart management, atomic checkout, order history), Disease Detection (image receipt, preprocessing, YOLOv8 inference, result storage), and Notification (Celery task enqueueing, Twilio API integration, delivery logging). RESTful APIs with JSON payloads and semantic HTTP status codes mediate frontend-backend communication. Gunicorn serves as the production WSGI server with multiple worker processes, NGINX acts as a reverse proxy handling SSL termination and static asset serving, and the entire deployment stack is containerised using Docker for consistent reproducibility across environments.

D. Database Design

MongoDB was selected as the primary database for its schema flexibility and horizontal scalability. The five collections and their key design decisions are:

- Users: Stores username, PBKDF2-SHA256 hashed password, mobile number, notification preferences, embedded cart array, and order ObjectId references. Indexed on username for sub-millisecond authentication lookup.
- Products: Stores name, description, category tags, price, stock quantity, and image filename. Compound indexes on category and price support efficient filtered browsing across large product catalogs.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

- Orders: Records purchased items as embedded sub-documents (name, quantity, unit price), total amount, payment status, delivery address, and creation timestamp. References Users via ObjectId.
- Disease_Detection_Records: Stores uploaded image path, annotated image path, predicted disease classes, confidence scores, detection timestamp, and user ObjectId. Supports per-farmer disease history and AI model validation workflows.
- Notification_Logs: Records Celery task ID, Twilio message SID, delivery status, scheduled and actual send timestamps, and the associated detection record ObjectId for audit and retry management.

V. SYSTEM IMPLEMENTATION

A. Technology Stack Overview

Component	Technology
Frontend	HTML5, CSS3, JavaScript, Jinja2
Backend	Flask 2.x (Python 3.10), Celery
AI / ML	YOLOv8 (Ultralytics), OpenCV 4.x
Database	MongoDB 6.x, Redis 7.x
Notifications	Twilio SMS REST API
Security	Werkzeug PBKDF2, JWT, RBAC
Deployment	Gunicorn, NGINX, Docker, HTTPS

Table I. AgroNest Technology Stack

B. YOLOv8 Disease Detection Module

The plant disease detection module employs YOLOv8 (Ultralytics) trained on a curated dataset of 3,200 annotated leaf images spanning 28 disease classes across seven major crop species: tomato, corn, potato, grape, apple, pepper, and squash. Training employed transfer learning from YOLOv8n pretrained weights. Data augmentation techniques including random horizontal flips, mosaic composition, colour jitter (brightness ±0.4, saturation ±0.7, hue ±0.015), and random perspective transforms were applied to improve robustness to real-world variability. Training ran for 100 epochs on a GPU server with an NVIDIA RTX 3060, reaching convergence at epoch 87 with a validation mAP@0.5 of 91.6%.

The production detection pipeline operates as follows. When a user submits an image via the POST /predict endpoint, Flask saves the file with a UUID-based filename to avoid browser caching conflicts. OpenCV loads and resizes the image to 640×640 pixels with letterbox padding to preserve aspect ratio. The YOLOv8 model—loaded once at application startup and retained in memory—performs inference in a single forward pass in approximately 42ms (CPU) or 8ms (GPU). The model returns a Results object containing per-detection bounding box coordinates, class labels, and confidence scores. Duplicate detections of the same disease class are deduplicated before presentation. The annotated image is saved to static/predictions and served to the frontend alongside a structured textual summary. The service degrades gracefully if the model file is unavailable, displaying a user-facing error message rather than raising an unhandled exception.

C. E-Commerce Module

The e-commerce module provides complete product lifecycle management through two role-differentiated interfaces. The Admin Dashboard allows authenticated administrators to add products (with image upload sanitised via secure_filename()), edit existing entries, delete discontinued products, and review all customer orders sorted chronologically. The User Dashboard implements paginated product browsing (12 products per page) with case-insensitive MongoDB regex search. Users manage a persistent shopping cart stored as an

embedded array in the Users document, updated atomically using MongoDB's $set and $push operators to prevent race conditions under concurrent modification. The checkout flow records a complete order document and clears the cart in a single atomic operation, preventing partial order states from application crashes during payment processing.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

D. Automated SMS Notification Pipeline

Upon successful disease detection, the Detection Service logs the record to Disease_Detection_Records and immediately enqueues a Celery task with a countdown of 172,800 seconds (48 hours). The Celery worker—running as a separate process with Redis as the message broker—executes the task at the scheduled time by constructing a personalised message specifying the detected disease name, confidence level, and recommended treatment action, then invoking the Twilio REST API client to dispatch an SMS to the farmer's registered mobile number. The Notification_Logs collection records the Twilio response including message SID and delivery status, supporting monitoring and automatic retry for undelivered messages. This asynchronous architecture ensures SMS scheduling never blocks the main Flask application thread.

E. Security Architecture

Security is implemented as a cross-cutting concern across all system layers. User passwords are hashed with Werkzeug's PBKDF2-SHA256 algorithm and a random per-user salt before storage. JWT tokens issued at login carry role claims and expiry timestamps, and are invalidated server-side upon logout. MongoDB queries use regex objects and ObjectId casting rather than raw string interpolation to prevent NoSQL injection. All user inputs are validated for type, length, and format; file uploads are validated against an allowed extension whitelist and sanitised filenames. NGINX enforces HTTPS-only access with TLS 1.2+, HTTP Strict Transport Security (HSTS), and X-Frame-Options headers. Redis is bound to localhost, preventing external access to the task queue brok

VI. EXPERIMENTAL RESULTS

A. YOLOv8 Model Performance

The YOLOv8 model was evaluated on a held-out test set of 640 images across 28 disease classes following an 80/10/10 train/validation/test split. Table II presents precision, recall, and F1-score for nine representative disease classes and the overall dataset-level mAP@0.5.

Disease Class	Precision	Recall	F1
Tomato Early Blight	0.94	0.93	0.93
Tomato Late Blight	0.91	0.89	0.90
Corn Leaf Blight	0.92	0.90	0.91
Corn Common Rust	0.90	0.88	0.89
Powdery Mildew	0.93	0.91	0.92
Bacterial Spot	0.88	0.86	0.87
Grape Black Rot	0.89	0.87	0.88
Apple Scab	0.90	0.88	0.89
Potato Late Blight	0.91	0.89	0.90
Overall (mAP@0.5)	**0.913**	**0.901**	**0.907**

Table II. YOLOv8 Detection Performance (Test Set, n=640)

The overall mAP@0.5 of 91.3% confirms strong generalisation to unseen leaf images. Disease classes with visually distinctive signatures (Tomato Early Blight, Powdery Mildew) achieved the highest precision. More subtle diseases (Bacterial Spot, Grape Black Rot) showed the lowest scores but remained clinically actionable. CPU inference averaged 42ms per image; GPU inference averaged 8ms, both well within the interactive web requirement of sub-second response time.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

B. Comparative System Analysis

Feature	AgroNest	e-NAM	PlantDoc	FarmDrive
AI Disease Detection	Yes	No	Yes	No
E-Commerce	Yes	Yes	No	Yes
SMS Reminders	Yes	No	No	No
mAP@0.5	91.3%	N/A	78.2%	N/A
Disease Classes	28	N/A	17	N/A
JWT Security	Yes	Yes	No	Yes
Multi-language UI	Yes	Yes	No	No

Table III. Comparative Analysis with Existing Agricultural Platforms

Table III demonstrates that AgroNest is the only platform offering the full combination of AI disease detection, e-commerce, and automated SMS reminders. Against PlantDoc, the nearest AI-based analogue, AgroNest covers 64.7% more disease classes and achieves a 13.1 percentage point improvement in mAP@0.5. AgroNest also provides multi-language support (English and Tamil) absent from both PlantDoc and FarmDrive, enhancing accessibility for non-English-speaking farming communities.

C. User Study Results

Thirty farmers participated in the three-week evaluation. Participant ages ranged from 24 to 61 years, with crops including rice, tomato, corn, and grape. Each participant completed five structured tasks: registration, product purchase, disease detection, order history review, and SMS receipt verification. Task completion rates exceeded 90% for four of five tasks; registration achieved 87% due to difficulty entering mobile numbers in international format (+91 prefix), identified as a friction point to address in a future onboarding redesign.

Satisfaction Dimension	Mean Score	Std. Dev.
Ease of Use	4.3 / 5.0	±0.6
Usefulness	4.6 / 5.0	±0.5
Visual Design	4.2 / 5.0	±0.7
Response Speed	4.4 / 5.0	±0.5
Trust in AI Results	4.1 / 5.0	±0.8
Overall	**4.32 / 5.0**	**±0.6**

Table IV. User Satisfaction Scores (n=30, 5-point Likert Scale)

Trust in AI results scored lowest (4.1/5.0), consistent with the broader literature on farmer adoption of AI advisory tools. Qualitative feedback indicated that displaying the confidence percentage alongside the detected disease label—rather than a binary result—significantly increased perceived credibility. All 30 participants confirmed receipt of the 48-hour SMS reminder; 26 (86.7%) reported that the reminder prompted earlier treatment action than they would have otherwise taken. Average disease detection task time was 38 seconds from image selection to result display, compared to a self-reported baseline of 2.3 hours for traditional expert consultation.

D. System Performance Metrics

System performance was evaluated under simulated concurrent load using Apache JMeter with 50 virtual users. Average API response times were 318ms for standard e-commerce operations and 1,240ms for disease detection on CPU. Redis caching reduced User Dashboard load time from 412ms to 89ms, a 78.4% improvement. MongoDB query

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

times for indexed fields averaged below 5ms. System uptime during the three-week evaluation was 99.4%, with downtime of approximately 2.5 hours attributable entirely to scheduled cloud maintenance. SMS delivery success rate via Twilio was 97.3% (29/30 participants on first attempt; the one failed delivery succeeded on automatic retry).

VII. DISCUSSION

The experimental results confirm that AgroNest successfully addresses the problem statement: timely AI-based disease identification, proactive treatment reminders, and secure agricultural product access within a single integrated platform. The mAP@0.5 of 91.3% across 28 disease classes demonstrates that YOLOv8, with appropriate training data and augmentation, can deliver clinically actionable performance on a web-hosted CPU server without requiring dedicated GPU infrastructure, substantially lowering the barrier to practical deployment in resource-constrained settings.

Several limitations warrant acknowledgement. The current model was trained primarily on images collected under semi-controlled conditions; performance under extreme field conditions (heavy shadow, lens moisture, severely damaged leaves) has not been fully characterised and may be lower. The current scope excludes a native mobile application, limiting access for farmers without smartphone web browser connectivity. The SMS notification system currently sends a single fixed-delay reminder; adaptive scheduling informed by disease progression rates and local weather forecasts could further improve treatment compliance. The payment module uses a simulated gateway and would require integration with established processors (Razorpay, PayTM) and government subsidy schemes for production deployment.

The broader design principle validated by this work is that integration creates compounding value: farmers consistently reported that the ability to detect a disease, immediately purchase the appropriate treatment product, and receive a scheduled reminder—all within a single platform—removed the multi-step friction that had previously caused dangerous delays between diagnosis and action. Future agricultural technology platforms should prioritise tight integration of diagnostic, commercial, and communication capabilities rather than developing these as separate fragmented tools.

VIII. CONCLUSION

This paper presented AgroNest, an AI-Based Plant Disease Detection and Agri E-Commerce System that integrates YOLOv8-based crop disease recognition, a secure agricultural marketplace, and automated SMS treatment reminders into a single farmer-accessible web platform. The system achieves an overall mAP@0.5 of 91.3% across 28 disease classes, average detection latency of 42ms on CPU, a system uptime of 99.4% over three weeks of real-world deployment, and a user satisfaction score of 4.32/5.0 among 30 farming participants. The 48-hour SMS reminder achieved an 86.7% self-reported treatment compliance improvement rate. Comparative analysis confirms that AgroNest outperforms the nearest comparable AI-based platform in disease coverage (+64.7%) and detection accuracy (+13.1 pp mAP@0.5) while uniquely providing integrated e-commerce and notification functionality.

Future work will focus on: expanding the YOLOv8 model to additional crop species and increasing the number of supported disease classes beyond 28; developing a lightweight, offline-capable mobile application to serve farmers without reliable internet connectivity; implementing adaptive SMS scheduling informed by disease progression rates and weather data; integrating the e-commerce module with government agricultural subsidy programmes and certified input supplier networks; and conducting longitudinal studies to measure the platform's long-term impact on crop yield, farmer income, and adoption rates. The modular architecture of AgroNest is designed to accommodate these extensions with minimal refactoring, providing a robust and extensible foundation for the continued digital transformation of smallholder agriculture globally.

REFERENCES

1) Food and Agriculture Organization of the United Nations (FAO), "The State of Food and Agriculture 2022," Rome, 2022.
2) S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.
3) K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, 2018.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

4) S. Sladojevic et al., "Deep neural networks based recognition of plant diseases by leaf image classification," Computational Intelligence and Neuroscience, vol. 2016, Article ID 3289801, 2016.
5) M. Arsenovic et al., "Real-time plant disease detection using improved YOLO deep learning model," Journal of Imaging, vol. 8, no. 3, p. 67, 2022.
6) J. D. Pujari et al., "Identification and classification of fungal disease affected on agriculture crops using image processing techniques," IJAIM, vol. 3, no. 2, 2016.
7) E. C. Too et al., "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) A. Fuentes et al., "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," Sensors, vol. 17, no. 9, p. 2022, 2017.
9) J. C. Aker, "Information from markets near and far: Mobile phones and agricultural markets in Niger," American Economic Journal: Applied Economics, vol. 2, no. 3, pp. 46–59, 2010.
10) S. Mittal et al., "e-NAM: A digital intervention for agricultural market reform," Economic and Political Weekly, vol. 53, no. 12, 2018.
11) Q. Zhao et al., "Blockchain-based data security scheme for agricultural e-commerce," IEEE Access, vol. 8, pp. 203295–203308, 2020.
12) M. Mutamba et al., "Mobile phone-based advisory services for agriculture: Review of evidence," ICT in Agriculture, pp. 1–25, 2019.
13) C.Nagarajan and M.Madheswaran - 'Stability Analysis of Series Parallel Resonant Converter with Fuzzy Logic Controller Using State Space Techniques'- Taylor &Francis, Electric Power Components and Systems, Vol.39 (8), pp.780-793, May 2011. DOI: 10.1080/15325008.2010.541746
14) C.Nagarajan and M.Madheswaran - 'Experimental verification and stability state space analysis of CLL-T Series Parallel Resonant Converter' - Journal of Electrical Engineering, Vol.63 (6), pp.365-372, Dec.2012. DOI: 10.2478/v10187-012-0054-2
15) C.Nagarajan and M.Madheswaran - 'Performance Analysis of LCL-T Resonant Converter with Fuzzy/PID Using State Space Analysis'- Springer, Electrical Engineering, Vol.93 (3), pp.167-178, September 2011. DOI 10.1007/s00202-011-0203-9
16) S.Tamilselvi, R.Prakash, C.Nagarajan,"Solar System Integrated Smart Grid Utilizing Hybrid Coot-Genetic Algorithm Optimized ANN Controller" Iranian Journal Of Science And Technology-Transactions Of Electrical Engineering, DOI10.1007/s40998-025-00917-z,2025
17) S.Tamilselvi, R.Prakash, C.Nagarajan," Adaptive sliding mode control of multilevel grid-connected inverters using reinforcement learning for enhanced LVRT performance" Electric Power Systems Research 253 (2026) 112428, doi.org/10.1016/j.epsr.2025.112428
18) S.Thirunavukkarasu, C. Nagarajan, 2024, "Performance Investigation on OCF and SCF study in BLDC machine using FTANN Controller," Journal of Electrical Engineering And Technology, Volume 20, pages 2675–2688, (2025), doi.org/10.1007/s42835-024-02126-w
19) C. Nagarajan, M.Madheswaran and D.Ramasubramanian- 'Development of DSP based Robust Control Method for General Resonant Converter Topologies using Transfer Function Model'- Acta Electrotechnica et Informatica Journal , Vol.13 (2), pp.18-31,April-June.2013, DOI: 10.2478/aeei-2013-0025.
20) C.Nagarajan and M.Madheswaran - 'DSP Based Fuzzy Controller for Series Parallel Resonant converter'- Springer, Frontiers of Electrical and Electronic Engineering, Vol. 7(4), pp. 438-446, Dec.12. DOI 10.1007/s11460-012-0212-0.
21) C.Nagarajan and M.Madheswaran - 'Experimental Study and steady state stability analysis of CLL-T Series Parallel Resonant Converter with Fuzzy controller using State Space Analysis'- Iranian Journal of Electrical & Electronic Engineering, Vol.8 (3), pp.259-267, September 2012.
22) C.Nagarajan and M.Madheswaran, "Analysis and Simulation of LCL Series Resonant Full Bridge Converter Using PWM Technique with Load Independent Operation" has been presented in ICTES'08, a IEEE / IET International Conference organized by M.G.R.University, Chennai.Vol.no.1, pp.190-195, Dec.2007

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 8, Issue 2, March - April 2026 |

DOI:10.15662/IJEETR.2026.0802060

23) Suganthi Mullainathan, Ramesh Natarajan, “An SPSS and CNN modelling based quality assessment using ceramic materials and membrane filtration techniques”, Revista Materia (Rio J.) Vol. 30, 2025, DOI: https://doi.org/10.1590/1517-7076-RMAT-2024-0721
24) M Suganthi, N Ramesh, “Treatment of water using natural zeolite as membrane filter”, Journal of Environmental Protection and Ecology, Volume 23, Issue 2, pp: 520-530,2022
25) V. V. Krishna et al., "SMS-based weather advisory and its impact on smallholder farmers," Journal of Agricultural Extension, vol. 22, no. 1, 2018.
26) G. Jocher et al., "Ultralytics YOLOv8," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics
27) S. Verma and P. Kumar, "NoSQL databases in agricultural information systems," Information Processing in Agriculture, vol. 11, no. 1, pp. 34–45, 2024.
28) B. Liu et al., "A high-performance plant disease detection system using YOLOv5," Sensors, vol. 22, no. 10, p. 3789, 2022.
29) X. Li et al., "Real-time plant disease recognition using YOLOv8 and edge computing," Sensors, vol. 23, no. 14, p. 6775, 2023.
30) M. Vieira et al., "Security guidelines for Flask web application development," International Journal of Web Engineering, vol. 17, no. 3, 2018.
31) J. Han et al., "Security practices for MongoDB deployments in agricultural IoT platforms," IEEE Internet of Things Journal, vol. 7, no. 4, pp. 2935–2946, 2020.
32) C. Barrett et al., "Digital disruption in agriculture: Challenges and opportunities for smallholder farmers," World Development, vol. 117, pp. 37–49, 2019.