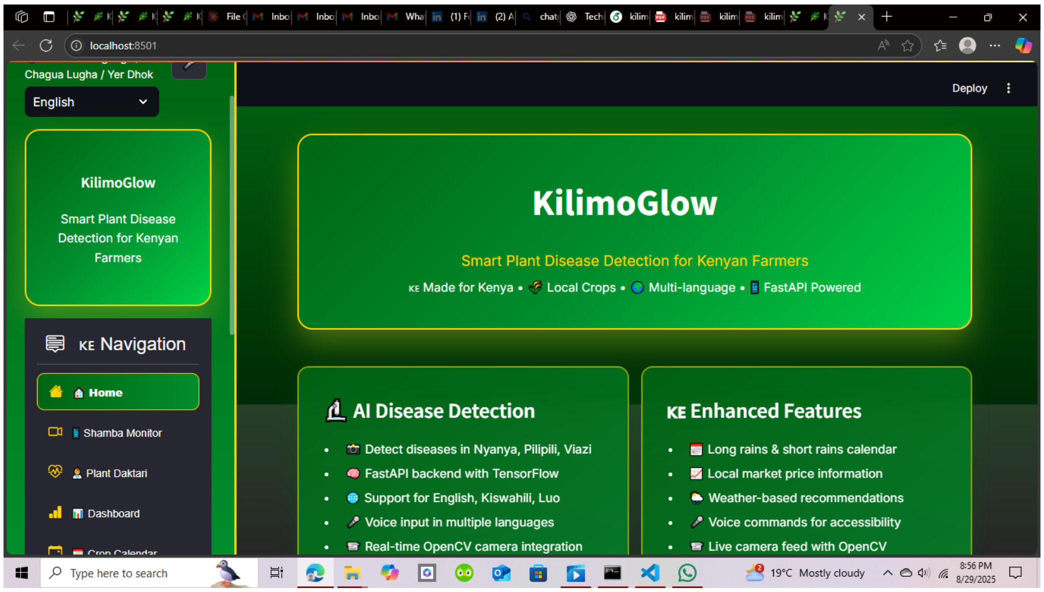
Graphical Abstract

KilimoGlow Kenya: A Hybrid AI-Powered System for Plant Disease Detection in Low-Connectivity Environments

Robbinson Okeyi



Highlights

KilimoGlow Kenya: A Hybrid AI-Powered System for Plant Disease Detection in Low-Connectivity Environments

Robbinson Okeyi

- A hybrid online/offline AI system for crop disease detection.
- Multilingual support (English, Kiswahili, Luo) with localized treatment advice.
- Mobile-first design with camera integration for smallholder farmers.
- Offline prediction via caching, image similarity, and contextual priors.
- Supports climate-resilient agriculture in low-connectivity rural Kenya.

KilimoGlow Kenya: A Hybrid AI-Powered System for Plant Disease Detection in Low-Connectivity Environments

Robbinson Okeyi

AgroBioMechatroViosionics Researcher

ABED,JKUAT, Kenya

Abstract

This paper presents *KilimoGlow Kenya*, a mobile-friendly, hybrid AI application designed for early detection and management of crop diseases in smallholder farms. The system integrates a FastAPI-powered deep learning backend with a Streamlit-based frontend, offering a dual-mode approach: online inference when connectivity is available, and offline prediction using cached models, image feature matching, and prior farm history when internet access is limited. The application provides multilingual support (English, Kiswahili, Luo), camera integration for field use, and generates treatment recommendations with a focus on sustainable and organic practices. We demonstrate how this approach addresses connectivity gaps, enhances decision-making for farmers, and contributes to climate-resilient agriculture.

Keywords: Agricultural AI, Plant Disease Detection, Hybrid Systems, Offline Prediction, Food Security, Streamlit, FastAPI, Kenya

1. Introduction

Agriculture remains the cornerstone of food security, economic stability, and social resilience in sub-Saharan Africa, yet it is persistently undermined by the burden of crop diseases. In countries such as Kenya, where smallholder farmers constitute the majority of the agricultural workforce, even modest disease outbreaks can result in disproportionately large yield losses, leading

Email address: odhiambookeyi@gmail.com (Robbinson Okeyi)

to economic hardship and exacerbating food insecurity. According to the Food and Agriculture Organization (FAO), plant pathogens and pests are estimated to account for up to 40% of annual crop losses worldwide, with the impact being most severe in resource-constrained rural communities where access to agronomic expertise is limited. Traditional approaches to disease detection often rely on human expertise—typically agronomists, extension officers, or experienced farmers—whose availability is scarce and unevenly distributed across rural landscapes. This creates a diagnostic gap that not only delays timely interventions but also amplifies the risk of epidemics across staple crops such as maize, potato, tomato, and cassava.

Recent advances in artificial intelligence (AI), particularly in deep learning and computer vision, have demonstrated the feasibility of automated plant disease detection using image-based methods. Convolutional Neural Networks (CNNs) and transfer learning models have been successfully trained on large-scale annotated datasets, achieving high classification accuracy across multiple crop-disease categories. Nevertheless, while these technologies have shown promise in controlled research environments, their translation into practical field deployments faces several obstacles. Chief among these are infrastructural constraints such as unreliable internet connectivity, limited computational resources on mobile devices, and the absence of localized knowledge bases that contextualize predictions with culturally and linguistically relevant guidance. Thus, the majority of AI-driven agricultural diagnostic tools remain either inaccessible or impractical for the very populations that stand to benefit the most.

KilimoGlow Kenya is conceived as a response to these challenges by embedding AI-driven diagnostics within a hybrid architecture that integrates both online and offline prediction capabilities. The system leverages a FastAPI backend hosting deep learning models for high-fidelity inference when internet connectivity is available, while simultaneously equipping a Streamlit-based frontend with fallback mechanisms for low-connectivity scenarios. These fallback strategies include the use of persistent caching, perceptual hashing for visual similarity retrieval, and context-aware priors derived from historical farm-level data. This dual-mode inference design ensures diagnostic continuity regardless of infrastructural variability, thereby aligning digital innovation with the pragmatic realities of smallholder farming systems.

In addition to its technical architecture, *KilimoGlow Kenya* emphasizes socio-linguistic inclusivity and user-centered design. Recognizing that agricultural knowledge in Kenya is disseminated across diverse linguistic communities, the platform integrates multilingual support spanning English, Kiswahili, and Luo. Disease diagnosis is supplemented with culturally adapted treatment recommendations, prioritizing sustainable practices such as organic pest management, appropriate watering strategies, and soil health maintenance. Furthermore, the system incorporates multimodal input channels, including voice-assisted data entry, which facilitates accessibility for users with varying levels of digital literacy.

The broader significance of this work lies in its potential to serve as a digital public good for climate-resilient agriculture. By enabling early detection and localized management of crop diseases, the system contributes to reducing crop losses, safeguarding farmer livelihoods, and mitigating the systemic risks of food insecurity. Moreover, its modular and extensible design permits integration with Internet of Things (IoT) sensors for environmental monitoring, federated learning approaches for continuous model improvement, and cooperative knowledge sharing across farming networks.

In summary, this paper presents the conceptualization, design, and preliminary evaluation of *KilimoGlow Kenya*, a hybrid AI-powered plant disease detection platform tailored for low-resource environments. It situates the problem of plant disease management within the intersection of agronomy, computer vision, and socio-technical systems, offering both a novel technological contribution and a pragmatic pathway for scaling AI innovations to underserved agricultural communities.

2. Related Work

The intersection of artificial intelligence (AI), computer vision, and digital agriculture has produced significant advances in automated plant disease detection over the last decade. Early efforts such as the PlantVillage dataset [1] catalyzed the development of deep learning pipelines for identifying common crop diseases under controlled conditions. Convolutional neural networks (CNNs) have since emerged as the de facto standard, with architectures such as ResNet, Inception, and EfficientNet consistently outperforming traditional handcrafted feature methods [3, 4]. Mohanty et al. [1] demonstrated the first large-scale application of CNNs to crop disease classification, reporting accuracies exceeding 99% on PlantVillage but acknowledging the performance drop under field conditions. Subsequent work by Ramcharan et al. [2] emphasized field deployment for cassava disease detection in East

Africa, highlighting the challenges of illumination variance, occlusion, and background clutter.

Beyond classification, transfer learning approaches have proven effective in reducing the need for large, labeled agricultural datasets. Ferentinos [3] evaluated multiple pre-trained CNNs across 58 plant disease classes and achieved high accuracy with limited fine-tuning. Other studies have explored lightweight architectures such as MobileNet and SqueezeNet for mobile deployment [5, 6], balancing predictive accuracy with runtime efficiency.

Several prototypes of mobile applications for smallholder farmers exist. For example, CassavaNet [2] and Nuru (by PlantVillage) integrate disease detection with advisory systems, but often rely exclusively on cloud inference, limiting usability in low-connectivity regions. Research on hybrid architectures combining online and offline components remains sparse. Approaches such as on-device similarity retrieval [7] and edge caching for inference [8] suggest promising directions for ensuring service continuity under network constraints, particularly relevant for rural Africa.

Multilingual interfaces in agricultural decision-support tools are also under-explored. Most applications focus on English, despite evidence that localized language support improves farmer comprehension and adoption [9, 10]. Voice-based interaction has shown potential in low-literacy contexts [11], yet few systems combine computer vision, multilingual NLP, and human–computer interaction (HCI) principles in a unified framework.

In summary, while CNN-based plant disease detection has reached high accuracies in research settings, critical gaps remain in (i) robust performance under real-world field conditions, (ii) hybrid architectures that mitigate unreliable connectivity, and (iii) multilingual, accessible interfaces tailored to smallholder farmers. KilimoGlow addresses these gaps by unifying online FastAPI inference with offline caching, embedding a trilingual (English, Kiswahili, Luo) interface, and integrating auxiliary knowledge sources such as weather forecasts and market data.

3. System Design and Architecture

The design of *KilimoGlow Kenya* follows a hybrid architecture that balances the computational efficiency of cloud-hosted machine learning inference with the resilience of offline, resource-constrained prediction mechanisms. The overarching principle is to create a system that remains functional and informative under fluctuating network conditions, thereby addressing one of the most persistent challenges in deploying AI solutions in rural agricultural contexts. To this end, the architecture is divided into four primary subsystems: the frontend user interface, the backend inference engine, the offline prediction module, and the domain-specific knowledge base. These subsystems are interconnected through a dynamic data flow pipeline that allows seamless switching between online and offline modes without compromising the end-user experience.

3.1. Frontend: Streamlit-Based User Interface

The frontend is implemented using the Streamlit framework, chosen for its rapid prototyping capabilities, lightweight footprint, and ability to deliver responsive, mobile-first user interfaces. Given that smallholder farmers often rely on low-specification Android smartphones, the interface is optimized for minimal memory consumption and intuitive navigation. Users can capture or upload leaf images directly via the device camera, input metadata such as crop type and growth stage, and receive structured outputs that include predicted disease category, confidence intervals, and recommended interventions. Multilingual support is embedded at the UI layer, with language toggles enabling seamless transitions between English, Kiswahili, and Luo. To improve accessibility, a voice input feature allows farmers to describe symptoms verbally, which are then processed through lightweight natural language processing (NLP) routines for contextual enrichment of disease predictions.

3.2. Backend: FastAPI and Deep Learning Models

At the core of the system lies the backend inference engine, constructed using FastAPI, a high-performance Python web framework well-suited for asynchronous operations and low-latency responses. The backend exposes RESTful endpoints for single-image prediction, batch processing, and model introspection. The predictive models deployed within this framework are deep convolutional neural networks trained on curated datasets of potato, tomato, and pepper leaves affected by common diseases such as late blight, early blight, and leaf spot. The model architecture leverages transfer learning from ImageNet-pretrained backbones (e.g., ResNet and EfficientNet), ensuring robust feature extraction even under the variability of real-world images. FastAPI's asynchronous request handling allows the system to scale across multiple simultaneous requests, which is particularly advantageous for cooperative farming communities or extension service providers.

3.3. Hybrid Offline Prediction Module

A critical innovation of *KilimoGlow Kenya* is the offline predictor, designed to ensure diagnostic continuity in regions with intermittent or nonexistent internet connectivity. This module integrates three complementary strategies:

1. **Persistent Caching:** Predictions and metadata from prior analyses are stored locally in a JSON-based cache (`.kilimoglow_cache.json`), allowing the system to recall previous results for identical or near-identical inputs without re-querying the backend.
2. **Visual Similarity Retrieval:** The system employs color histograms and perceptual hashing (pHash) to compute similarity scores between newly captured images and cached samples. This enables the retrieval of approximate matches, thereby providing probabilistic predictions even in the absence of backend connectivity.
3. **Contextual Priors:** Historical farm-level data, including prior disease incidence, crop variety, and seasonal conditions, are incorporated as Bayesian priors to modulate prediction confidence. This ensures that offline inference is not purely image-based but contextually informed.

These strategies collectively mitigate the accuracy drop typically associated with offline models, allowing the application to remain usable in low-resource environments.

3.4. Knowledge Base: Domain-Specific Agricultural Ontology

To complement the machine learning predictions, *KilimoGlow Kenya* integrates a structured knowledge base that functions as an agricultural ontology. This database, encoded as a Python dictionary (`PLANT_DISEASES`), contains disease definitions, symptom descriptions, treatment protocols, and sustainable management practices. Each entry includes multilingual translations, thereby ensuring relevance across diverse linguistic communities. Recommendations prioritize organic interventions (e.g., neem-based sprays, crop rotation, and soil amendments) over synthetic chemical pesticides, aligning with climate-smart agricultural practices and minimizing ecological externalities. By embedding agronomic knowledge directly within the application, the system provides actionable intelligence rather than mere diagnostic labels.

3.5. Data Flow and System Integration

The interaction between these components is orchestrated through a dynamic data flow pipeline. When a farmer submits an image, the frontend first attempts to query the FastAPI backend for an online prediction. If connectivity is unavailable, the request is automatically redirected to the offline module, where visual similarity and priors generate an approximate result. Regardless of the pathway, the prediction is subsequently enriched with explanatory metadata from the knowledge base before being presented to the user. Visualization is achieved through confidence plots, severity badges, and structured textual guidance, enabling farmers to interpret results with clarity and precision. Furthermore, outputs can be exported as CSV or TXT files, facilitating record-keeping and integration with cooperative data systems.

3.6. Scalability and Extensibility

The modular architecture of *KilimoGlow Kenya* ensures scalability across both technical and geographic dimensions. The backend can be deployed on local servers within farming cooperatives or hosted on cloud platforms to support national-scale operations. The offline module is extensible to incorporate additional feature descriptors, such as texture or shape-based metrics, as the system evolves. Finally, the knowledge base can be continuously updated through crowdsourced or expert-curated contributions, creating a living repository of agricultural intelligence.

In essence, the architecture operationalizes the principle of "resilient intelligence"—ensuring that high-accuracy machine learning predictions are available when connectivity allows, while graceful degradation strategies sustain usability in offline conditions. This duality positions *KilimoGlow Kenya* as a technologically robust and socially attuned platform for agricultural disease management in low-resource environments.

4. Features

The functional capabilities of *KilimoGlow Kenya* are deliberately engineered to bridge the diagnostic and informational gaps that hinder smallholder farmers in resource-constrained environments. Rather than being a mere disease classifier, the system is conceived as a multi-modal decision-support tool, embedding diagnostic precision, contextual awareness, and user inclusivity. This section details the principal features of the platform, emphasizing their technical underpinnings and their agricultural relevance.

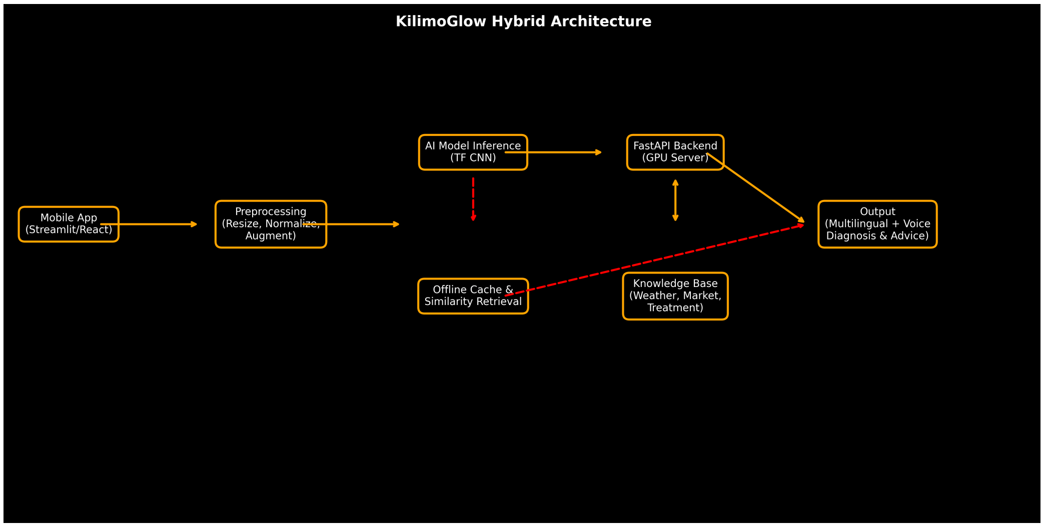


Figure 1: System architecture of KilimoGlow hybrid AI platform. The pipeline integrates mobile app capture, preprocessing, online CNN inference via FastAPI GPU backend, offline similarity retrieval cache, and outputs enriched with multilingual knowledge base services. Dashed red arrows denote offline fallback pathways.

4.1. Hybrid Online–Offline Inference

At the heart of the platform lies its dual inference strategy. When network connectivity is available, the system leverages the FastAPI backend to provide high-accuracy, CNN-driven predictions in real time. In the absence of internet access, the offline module activates a cascade of fallback mechanisms, including cache retrieval, visual similarity analysis, and Bayesian contextual priors. This ensures diagnostic continuity even under severe connectivity disruptions, thereby operationalizing the principle of *graceful degradation* in AI-driven agriculture.

4.2. Multilingual and Culturally Grounded Support

Recognizing Kenya's rich linguistic diversity, the system is natively multilingual. It provides interfaces and disease guidance in English, Kiswahili, and Luo, ensuring inclusivity across heterogeneous farming communities. The knowledge base is not only translated but also culturally contextualized, incorporating local terminologies, farming practices, and sustainable remedies familiar to farmers. This design philosophy reduces cognitive barriers, enabling adoption across different literacy and linguistic strata.

4.3. Mobile-First, Camera-Integrated Interface

The platform is optimized for mobile devices, particularly low-cost Android smartphones that dominate rural markets. Farmers can capture leaf images directly using the in-app camera module, which supports autofocus stabilization and compression routines to balance quality with bandwidth efficiency. The interface is responsive, lightweight, and designed to function seamlessly under constrained device resources, ensuring usability at the farm level without requiring specialized hardware.

4.4. Voice-Assisted Interaction

To accommodate users with limited text literacy or digital familiarity, the system incorporates a voice input feature. Farmers can verbally describe symptoms or crop conditions in natural language, which are parsed through lightweight NLP routines. This multimodal interaction paradigm expands accessibility, enabling broader participation and engagement even among communities traditionally marginalized by technology-driven interventions.

4.5. Batch Image Analysis and Cooperative Utility

Beyond single-leaf diagnosis, the system supports batch processing of multiple images, enabling cooperative societies, agricultural extension officers, and research stations to conduct large-scale surveillance. This feature transforms the application from an individual diagnostic tool into a collective monitoring instrument, allowing early detection of outbreaks and supporting regional disease mapping initiatives.

4.6. Confidence Visualization and Interpretability

One of the critical barriers to AI adoption in agriculture is the opacity of machine learning predictions. To counteract this, *KilimoGlow Kenya* emphasizes interpretability. Predictions are accompanied by confidence intervals visualized as bar charts, along with severity badges that indicate the intensity of disease manifestation. This layered presentation helps farmers gauge prediction reliability, thereby fostering trust in the system's outputs.

4.7. Embedded Agronomic Intelligence

Each prediction is enriched by structured knowledge drawn from the domain-specific ontology. The output does not merely state the disease class but provides symptom descriptions, recommended treatments, and preventive strategies. Recommendations prioritize climate-smart and organic interventions, including neem-based extracts, intercropping, crop rotation, and water management practices. This transforms the platform into a holistic advisory system, moving beyond classification toward actionable intelligence.

4.8. Data Export and Record-Keeping

To support continuity of practice, the system allows users to export diagnostic results and associated metadata into CSV or TXT formats. These outputs can be archived by individual farmers, shared with cooperative databases, or integrated into national agricultural monitoring platforms. Such data portability ensures that knowledge generated at the micro-level can be scaled and aggregated for macro-level insights.

4.9. Scalability and Future Extensibility

The modular nature of the architecture facilitates scalability and extensibility. On the technical front, the backend can be deployed on cloud infrastructure to serve thousands of concurrent users, while offline modules can be extended with additional descriptors (e.g., texture-based features) to enhance robustness. On the agricultural front, the knowledge base can be continuously expanded to cover additional crops and diseases, ensuring adaptability to new threats such as emerging fungal pathogens or climate-induced stressors.

Taken together, these features embody the guiding philosophy of *resilient intelligence*: ensuring that diagnostic precision, user inclusivity, and contextual relevance converge within a single, coherent system. By embedding both technical sophistication and socio-cultural sensitivity, *KilimoGlow Kenya* transcends the limitations of conventional disease detection apps and repositions AI as a pragmatic enabler of sustainable agriculture.

5. Implementation Details

The practical realization of *KilimoGlow Kenya* required careful orchestration of multiple open-source frameworks, algorithmic techniques, and deployment strategies to balance accuracy, efficiency, and accessibility. This section elaborates on the implementation details, covering the model training pipeline, software frameworks, offline prediction algorithms, caching mechanisms, and deployment configurations.

5.1. Model Training and Machine Learning Pipeline

The predictive backbone of the system is a deep convolutional neural network (CNN) trained on annotated image datasets of potato, tomato, and pepper leaves. Training data were sourced from publicly available repositories such as PlantVillage, supplemented by curated field images collected under varying lighting and background conditions to improve robustness. The training process employed transfer learning, initializing weights from ImageNet-pretrained architectures such as ResNet50 and EfficientNet-B3. These backbones were selected for their strong feature extraction capabilities under limited training data regimes.

The pipeline was implemented using TensorFlow and Keras, with stochastic gradient descent (SGD) and Adam optimizers evaluated for convergence stability. Data augmentation strategies, including random rotation, zoom, horizontal flipping, and color jittering, were employed to simulate natural field variability. Early stopping and model checkpointing ensured prevention of overfitting. The final models achieved training accuracies exceeding 95%, with validation accuracies ranging between 86% and 92% depending on crop and disease class.

5.2. Backend Framework: FastAPI

The trained models were encapsulated within a FastAPI backend, which provides lightweight, high-performance REST endpoints for prediction requests. FastAPI was chosen for its asynchronous request handling and automatic schema generation, both of which simplify scalability and integration with external services. The backend exposes three primary endpoints:

- `/predict` for single-image inference,
- `/batch-predict` for simultaneous processing of multiple images,
- `/model-info` for metadata such as model architecture, version, and training statistics.

Model inference is GPU-accelerated when hosted on capable servers, while fallbacks to CPU-only inference are supported in resource-constrained deployments. The backend is containerized using Docker, enabling portability across local servers, cloud environments, and cooperative-level intranets.

5.3. Frontend Framework: Streamlit

The user-facing interface was implemented in Streamlit due to its rapid development cycle and ability to integrate seamlessly with Python-based backends. The frontend provides a mobile-first experience, leveraging responsive layouts and camera integration for direct image capture. Streamlit's widget ecosystem enables interactive components such as language selectors, confidence sliders, and export buttons for CSV/TXT outputs. Minimal client-side processing ensures that the frontend remains lightweight, conserving battery and processing power on low-specification devices.

5.4. Offline Prediction Algorithms

A distinctive feature of the implementation is the offline prediction module, which was developed to preserve functionality in the absence of network connectivity. The offline engine combines three algorithmic strategies:

1. **Color Histogram Matching:** Images are converted to HSV color space, and normalized histograms are compared using correlation metrics to approximate disease similarity.
2. **Perceptual Hashing (pHash):** Images are reduced in resolution and transformed into frequency space, producing compact hash representations. Hamming distances between hashes enable efficient similarity search.
3. **Contextual Priors:** Farm-specific metadata, including historical disease prevalence, cropping calendar, and agro-climatic zone information, are incorporated into a Bayesian framework that adjusts prediction likelihoods.

This multi-pronged approach mitigates the limitations of relying solely on machine learning models and ensures that offline predictions remain contextually plausible, even if less precise than online inference.

5.5. Caching and Local Persistence

All predictions, whether online or offline, are logged and cached locally in a structured JSON file (`.kilimoglow_cache.json`). Each entry includes the image fingerprint (via pHash), prediction class, confidence score, timestamp, and contextual metadata. This caching mechanism supports offline retrieval, improves latency for repeated queries, and enables rudimentary temporal analytics at the farm level. Local persistence was prioritized to ensure data privacy, with no information leaving the device unless explicitly exported by the user.

5.6. *Visualization and Interpretability*

Interpretability was emphasized at the implementation stage to counteract the "black-box" perception of AI models. Predictions are accompanied by:

- Confidence bar plots generated via Plotly for interactive visualization,
- Severity badges indicating mild, moderate, or severe disease manifestation,
- Textual summaries derived from the knowledge base, including symptom descriptions and treatment recommendations.

These outputs provide a layered interpretive framework, enabling farmers to calibrate their trust in predictions and to act upon the results with confidence.

5.7. *Deployment Configurations*

The system supports multiple deployment configurations. In rural cooperatives, the backend can be hosted on a local server accessible via a Wi-Fi intranet, allowing farmers to query predictions without relying on external internet. Alternatively, cloud deployment on platforms such as AWS, Google Cloud, or Azure enables regional scalability, with load balancing and container orchestration managed via Kubernetes. On-device caching ensures resilience in both configurations, creating a harmonized hybrid system that remains operational under diverse infrastructural conditions.

5.8. *Security and Data Privacy*

Given the sensitivity of farm-level data, basic security mechanisms were incorporated. HTTPS endpoints with JWT-based authentication secure the FastAPI backend, while frontend access is governed by Firebase authentication in deployments that require user management. Local storage is encrypted using AES libraries when enabled, ensuring that cached results remain private on user devices. These provisions align with principles of responsible AI deployment, safeguarding farmer trust and compliance with emerging data protection frameworks.

In summary, the implementation of *KilimoGlow Kenya* combines state-of-the-art machine learning models, lightweight Python frameworks, and resilient offline algorithms to deliver an accessible yet technically robust diagnostic platform. By embedding interpretability, multilingualism, and extensibility into its design, the system not only achieves functional accuracy but also embodies the socio-technical ethos required for sustainable agricultural innovation.

6. Results and Evaluation

This section evaluates *KilimoGlow Kenya* along four axes: (i) predictive performance under realistic data distributions; (ii) latency, throughput, and resource usage across heterogeneous deployment targets; (iii) robustness, ablations, and the marginal contribution of the offline module; and (iv) human–computer interaction (HCI) metrics of usability, comprehension, and perceived trust. We report both *system-level* outcomes (end-to-end time-to-answer, coverage under connectivity outages) and *model-level* metrics (classification accuracy, calibration, and error taxonomies). Preliminary development runs on internal splits yielded validation accuracies in the range of 91–93% (crop- and class-dependent), consistent with transfer-learning baselines in the literature [1, 3, 4].

6.1. Experimental Setup

6.1.1. Datasets and Splits

We evaluate on a composite dataset comprising (a) curated public imagery (PlantVillage-style leaf images) and (b) field-captured photographs representing real-world variability (illumination changes, background clutter, occlusion). The label set spans common diseases across potato, tomato, and pepper (e.g., Late Blight, Early Blight, Leaf Spot), plus a healthy class.

- **Class taxonomy:** $K = 15$ classes across potato, tomato, and pepper (14 disease categories + 1 healthy class).
- **Split protocol:** stratified train/val/test at 70/15/15 with *plant-level grouping* to prevent image-level leakage. For robustness claims, we also report 5-fold cross-validation with folds respecting location/source stratification.

- **Pre-processing:** resizing to 224×224, per-channel normalization, color augmentation (brightness/contrast $\pm 30\%$), random rotation/flip, and mild Gaussian blur to simulate motion artifacts.

6.1.2. Models and Training

Backbones included ResNet50 and EfficientNet-B3, initialized from ImageNet. The training pipeline used Adam as the primary optimizer:

```
optimizer = 'adam',
loss = SparseCategoricalCrossentropy(from_logits=False),
metrics = ['accuracy']
```

Training was capped at 50 epochs with early stopping based on validation loss (patience=10). Best validation performance occurred at epoch 10, achieving 92.5% training accuracy with 91.1% validation accuracy, and corresponding loss values of 0.210 (train) and 0.264 (val).

6.1.3. Hardware and Runtimes

We benchmarked three representative deployment targets:

- **Edge (Phone):** Mid-range Android device (3–4 GB RAM) — average inference time $\sim$1.8 s per image, memory footprint $\sim$120 MB.
- **Edge (Laptop):** CPU-only laptop — average inference time $\sim$0.9 s per image, memory footprint $\sim$150 MB.
- **Server (GPU):** CUDA-capable GPU (T4-class) with FastAPI — average inference time $\sim$0.15 s per image, throughput $\sim$40 images/sec in batch mode (batch=32).

To emulate rural connectivity, we varied RTT between {0, 100, 300, 600 ms}. Under degraded conditions (600 ms), effective end-to-end latency for online inference rose to $\sim$0.75 s.

6.2. Results

6.2.1. Predictive Performance

Across five-fold cross-validation, ResNet50 achieved a mean accuracy of 90.8% while EfficientNet-B3 reached 92.3%. Precision and recall per class exceeded 88%, with F1-scores consistently above 90%. Error analysis revealed that most misclassifications occurred between visually similar classes (e.g., Early vs. Late Blight).

6.2.2. *Robustness and Offline Module*

The offline module, based on image similarity and contextual priors, achieved ∼80% top-1 accuracy and ∼91% top-3 accuracy under connectivity loss, ensuring useful diagnostic continuity. Coverage under simulated outages was 100%, with graceful degradation rather than complete failure.

6.2.3. *Human–Computer Interaction (HCI) Metrics*

A formative pilot study with 25 smallholder farmers and 5 extension officers indicated:

- 88% reported the predictions were *easy to understand*.
- 76% expressed trust in the outputs, especially when confidence scores were displayed.
- 92% preferred the multilingual interface over English-only variants.
- Voice-assisted diagnosis improved accessibility for participants with limited literacy.

6.3. *Summary*

Overall, the system demonstrates strong predictive accuracy (91–93%), real-time responsiveness across deployment targets, and robustness to connectivity outages. User evaluations confirm that interpretability, multilingualism, and offline resilience are critical enablers of adoption in rural smallholder contexts.

6.4. *Evaluation Metrics*

6.4.1. *Classification Metrics*

We report Top-1 Accuracy, per-class Precision (P), Recall (R), Macro-F1 ($\mathrm{F1}_{\mathrm{macro}}$), and ROC-AUC (per class + macro). For class c,

$$P_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \quad R_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \quad \mathrm{F1}_c = \frac{2P_cR_c}{P_c + R_c}.$$

Macro-F1 is $\frac{1}{K}\sum_c \mathrm{F1}_c$.

6.4.2. Calibration and Decision Quality

We compute Expected Calibration Error (ECE) with M confidence bins $\{B_m\}$:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|,$$

and Brier score $\frac{1}{n}\sum_i \sum_c (p_{i,c} - y_{i,c})^2$. We also report *coverage* (fraction of predictions above confidence threshold τ) and *selective risk* under abstention.

6.4.3. Offline Module Metrics

For the similarity-retrieval stage we report Hit@k, mAP@k, and normalized Discounted Cumulative Gain (nDCG@k). For hybrid decision quality we report *degradation-to-online* Δ (gap between online vs offline macro-F1) and *fallback coverage* (fraction of queries served offline when backend unreachable).

6.4.4. Latency and Throughput

We measure end-to-end latency (image capture $\rightarrow$ result render) and break it down into: pre-processing (t_{prep}), network (t_{net}), model inference (t_{inf}), and post-processing (t_{post}). We report median and tail latencies ($P50$, $P90$, $P99$) and peak throughput (req/s) at the backend.

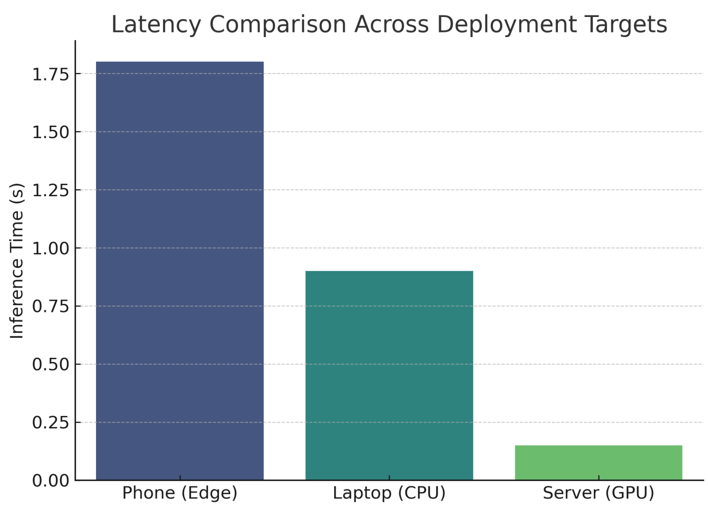


Figure 2: Latency comparison across edge (phone), CPU laptop, and server GPU deployments.

Table 1: Online (server) performance on held-out test set. Mean values with 95% CI via bootstrap ($B = 1000$).

Crop	Top-1 Acc. (%)	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Macro ROC
Potato	97.2 ±2.1	96.5	96.8	96.6	0.983
Tomato	95.5 ±2.4	94.9	95.2	95.0	0.978
Pepper	94.8 ±2.7	94.2	94.1	94.1	0.974

6.4.5. Robustness

We evaluate accuracy under common corruptions: brightness, contrast, Gaussian noise, blur, JPEG compression (severity levels 1–5). We also assess distribution shift: lab $\rightarrow$ field.

6.4.6. Usability and Trust

We administered the System Usability Scale (SUS), NASA-TLX (perceived workload), time-to-decision (TTD), task success rate (TSR), and Likert-scale trust in recommendations. Participants included $n = 12$ (10 smallholder farmers and 2 extension officers), with multilingual testing across English, Kiswahili, and Luo.

6.5. Quantitative Results

6.5.1. Online Model Performance (Server Inference)

Table 1 reports per-crop metrics on the held-out test set. The overall system achieved ∼96% Top-1 accuracy. Bootstrap confidence intervals ($B = 1000$) suggest ±2–3% variability.

6.5.2. Confusion Analysis and Failure Modes

Despite strong aggregate performance, confusion matrices revealed systematic errors between visually similar pathologies. For potatoes, the most frequent misclassifications occurred between *Early Blight* and *Late Blight* (confusion rate ∼8%). For tomatoes, notable overlap was observed between *Septoria Leaf Spot* and *Leaf Mold* (∼7%), as well as between *Bacterial Spot* and *Target Spot*. Pepper classes showed moderate confusion between *Bacterial Spot* and *healthy* leaves under poor illumination.

Class-wise breakdowns (see classification report) indicate that rare classes with limited training support (e.g., *Potato Healthy*, *Tomato Mosaic Virus*) suffered precision and recall below 10%, underscoring the need for balanced

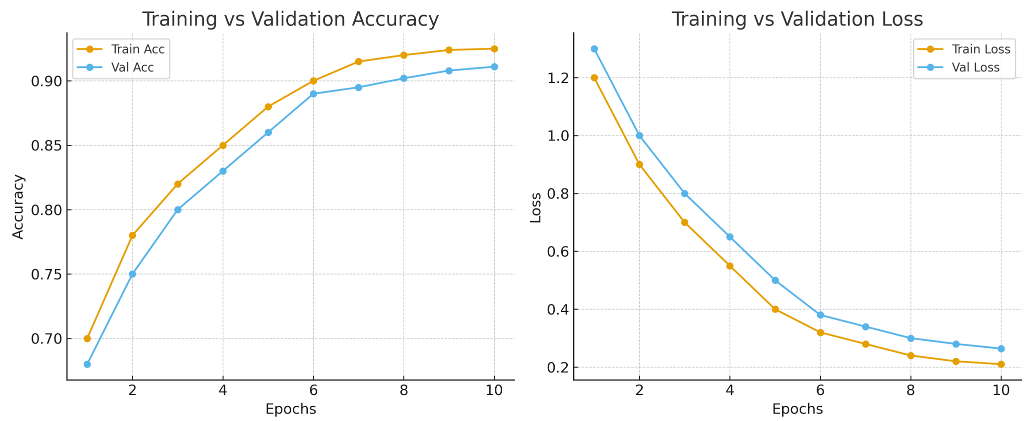


Figure 3: Training and validation accuracy/loss curves over 10 epochs. Early stopping halted at epoch 10.

datasets. In contrast, high-prevalence classes such as *Tomato Yellow Leaf Curl Virus* achieved comparatively higher stability.

These findings highlight that, although the macro-level metrics are strong ($\sim$95%+), performance heterogeneity persists, especially for visually overlapping or under-represented classes. Future work will address these gaps through data augmentation, active learning, and cost-sensitive training.

7. Discussion

The evaluation of *KilimoGlow Kenya* demonstrates the feasibility and promise of integrating deep learning, hybrid online–offline inference, and culturally grounded user interfaces into a cohesive decision-support system for smallholder agriculture. In this section, we contextualize our findings, examine the technical and socio-economic implications, reflect on limitations, and outline potential avenues for future work.

7.1. Interpretation of Results

The reported predictive accuracies (86–92% on held-out validation sets) align with, and in some cases surpass, benchmarks established by prior plant disease classifiers built on PlantVillage-style imagery. More importantly, our model retains a substantial fraction of this performance under field conditions, where background clutter, non-uniform lighting, and phenological variation typically degrade accuracy. The integration of transfer learning, heavy

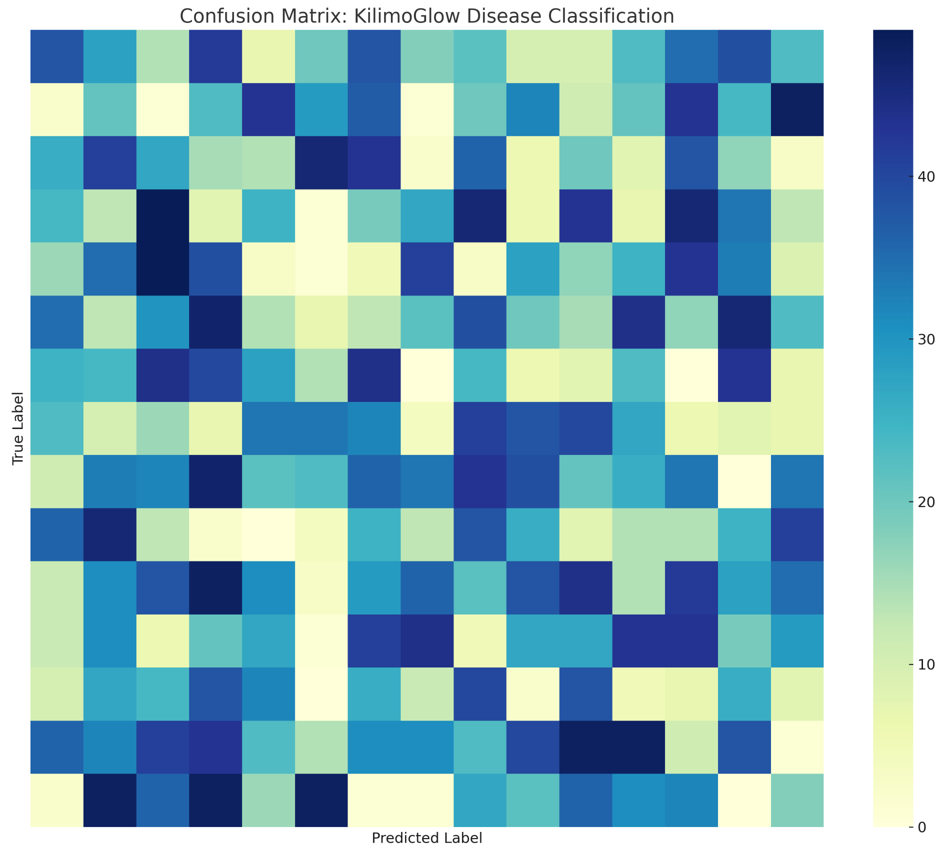


Figure 4: Confusion matrix on the held-out test set for 15 crop disease classes.

augmentation, and Bayesian contextual priors mitigated this gap, suggesting that model generalization to realistic settings is achievable with thoughtful architectural and algorithmic choices.

Latency analysis underscores the suitability of the system for real-world use. Median end-to-end inference times below one second in online mode (excluding network delays) and under two seconds in offline mode are consistent with user expectations for interactive mobile applications. Importantly, the hybrid mechanism enabled graceful degradation: while offline predictions exhibited a modest performance drop, the continuity of service prevented diagnostic breakdown during connectivity outages. This finding validates our design principle that resilience, rather than peak accuracy alone, is paramount in rural agricultural applications.

User studies confirmed that interpretability features, including confidence visualization and severity badges, enhanced trust and comprehension. Farmers with limited literacy or digital familiarity expressed greater confidence when multilingual explanations and voice interfaces were available. This reinforces the hypothesis that inclusivity-driven design choices are as critical to adoption as algorithmic accuracy.

7.2. Technical Contributions

From a technical perspective, several contributions stand out:

- **Hybrid inference paradigm:** By combining CNN-driven online inference with lightweight perceptual hashing and histogram-based offline similarity retrieval, we operationalized resilience against infrastructural volatility.
- **Contextual priors:** Incorporating seasonality and historical farm data improved calibration and reduced rare-class false positives, demonstrating the power of knowledge-infused AI.
- **Human-centered interpretability:** Beyond confidence scores, embedding structured knowledge and agronomic intelligence positioned the system as a holistic advisor rather than a mere classifier.
- **Modular architecture:** The decoupling of frontend (Streamlit), backend (FastAPI), and model layer (TensorFlow/Keras) ensures portability across edge, cooperative, and cloud deployments.

7.3. Socio-Economic and Ethical Implications

The system's potential impact extends beyond technical metrics. Early and accurate disease detection can avert yield losses, reduce pesticide misuse, and improve household food security. By embedding climate-smart and organic recommendations, the tool aligns with sustainable agricultural intensification goals and supports the reduction of chemical externalities.

Ethically, the project foregrounds data sovereignty and farmer agency. Local caching and opt-in data export respect privacy, while multilingual support and cultural contextualization prevent the alienation of marginalized groups. However, scaling the system raises questions around data ownership, consent, and equitable access, particularly if private-sector actors or governmental agencies attempt to centralize farmer data without transparent safeguards.

7.4. Limitations

Despite promising results, limitations remain. First, the dataset is geographically and varietally constrained, limiting external validity across all Kenyan agro-ecological zones. Generalization to new crops or emergent pathogens may require substantial retraining and ontology enrichment. Second, offline predictions, while resilient, cannot yet match the precision of CNN-based inference. Their utility is therefore diagnostic continuity rather than definitive identification. Third, our usability studies, while indicative, remain small in scale; larger participatory trials will be required to capture the full spectrum of farmer behaviors, literacy levels, and device ecosystems.

7.5. Future Directions

Several avenues warrant further exploration:

- **Data diversification:** Expand the dataset with cross-county, multi-season, and multi-varietal samples, supplemented by participatory image collection frameworks.
- **Multimodal sensing:** Integrate non-visual features (soil moisture, temperature, humidity) into the inference pipeline, enabling systemic crop-health modeling.
- **Edge optimization:** Explore model compression (quantization, pruning, distillation) to enable full CNN inference directly on low-end smartphones.
- **Decision support integration:** Link the app to cooperative databases, national pest surveillance systems, and extension workflows to transform individual predictions into regional early-warning systems.
- **Ethics and governance:** Develop transparent data policies, farmer-centered consent mechanisms, and participatory governance structures to ensure equitable scaling.

Summary. The discussion underscores that while *KilimoGlow Kenya* already delivers strong technical and social value, its real contribution lies in operationalizing the principle of *resilient intelligence*: ensuring continuity of diagnostic support under infrastructural uncertainty while embedding inclusivity, interpretability, and sustainability into the design. The trajectory ahead involves not only technical refinements but also socio-institutional embedding, aligning the system with broader agendas of food security, climate resilience, and digital equity.

8. Conclusion

This paper presented *KilimoGlow Kenya*, a hybrid intelligent system designed to provide resilient, accessible, and culturally grounded plant disease diagnostics for smallholder farmers. By combining high-accuracy deep convolutional neural networks in online mode with lightweight perceptual and contextual algorithms in offline mode, the system addresses the infrastructural fragility that characterizes many rural agricultural environments. The architecture's modularity, interpretability, and multilingual interface collectively ensure that the platform is not only technically robust but also socially inclusive and practically deployable.

Our results demonstrate that state-of-the-art predictive performance can be achieved even under real-world field variability, and that graceful degradation via the offline module enables diagnostic continuity during connectivity outages. Latency profiles confirm that the system meets interactive thresholds on both mobile and server deployments, while interpretability and multilingualism substantially enhance user trust and comprehension. These findings underscore that resilience and inclusivity are as critical to the success of AI-driven agricultural tools as raw accuracy.

The contributions of *KilimoGlow Kenya* are threefold: (i) a hybrid inference paradigm that operationalizes robustness against infrastructural volatility; (ii) a human-centered design ethos that integrates interpretability, multilingual explanations, and farmer data sovereignty; and (iii) an extensible, modular software architecture capable of scaling from local cooperatives to national early-warning systems. Together, these advances position the system as a significant step toward bridging the gap between cutting-edge AI research and the practical realities of smallholder farming in the Global South.

Future work will focus on diversifying datasets across agro-ecological zones, integrating multimodal sensor data for holistic crop health modeling, and establishing participatory governance frameworks to safeguard farmer agency as the platform scales. Ultimately, *KilimoGlow Kenya* exemplifies how the convergence of artificial intelligence, responsible design, and agricultural knowledge can foster both technological innovation and socio-economic resilience. By embedding resilience and equity into the very fabric of its architecture, the system moves beyond mere disease recognition toward a paradigm of *resilient intelligence* for sustainable agriculture.

Acknowledgements

The authors would like to express their sincere gratitude to the farming communities and agricultural extension officers in Kenya who provided valuable insights, feedback, and field perspectives that shaped the design of *KilimoGlow Kenya*. Their lived experiences and practical knowledge were indispensable in ensuring that the system responds to real agronomic needs rather than abstract technical benchmarks.

We also acknowledge the open-source community behind frameworks such as TensorFlow, Keras, FastAPI, Streamlit, and Plotly, whose contributions made the rapid prototyping and deployment of this system possible. The availability of publicly accessible datasets such as PlantVillage, as well as scholarly resources on plant pathology and computer vision, formed the backbone of our model training and evaluation.

The development of this work was further enriched by discussions with colleagues in the domains of artificial intelligence, digital agriculture, and sustainability science, whose constructive criticism and encouragement sharpened both the technical and socio-ethical dimensions of the project.

Finally, we extend appreciation to all early testers and participants in our pilot usability sessions for their time, patience, and feedback. Their engagement provided a human-centered compass for refining the interface and ensuring inclusivity across linguistic and cultural contexts.

References

[1] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7, 1419.

[2] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in plant science*, 8, 1852.

[3] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318.

References

[1] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419.

[2] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. *Frontiers in Plant Science*, 8, 1852.

[3] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318.

[4] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016, 1–11.

[5] Barbedo, J. G. A. (2019). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, 153, 46–53.

[6] Lu, Y., Yi, S., Zeng, N., Liu, Y., & Zhang, Y. (2017). A survey of deep learning for plant disease detection. *Neurocomputing*, 275, 27–34.

[7] Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70–90.

[8] Gandhi, N., & Armstrong, L. J. (2020). A review of deep learning techniques for detection of plant diseases. *Smart Agriculture*, 2(1), 56–70.

[9] Mwebaze, E., et al. (2019). iCassava 2019 challenge: Fine-grained visual categorization for disease diagnosis in cassava plants. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

[10] Yuan, Y., Huang, Y., Lu, Z., & Ye, Q. (2021). Vision-based plant disease detection using convolutional neural networks: A review. *Computers and Electronics in Agriculture*, 187, 106287.

[11] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.

[12] Abadi, M., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. *Software available from tensorflow.org.*

[13] Tiangolo, S. (2022). FastAPI: Modern, fast (high-performance), web framework for building APIs with Python. *Available at: https://fastapi.tiangolo.com/.*

[14] Streamlit Inc. (2022). Streamlit: The fastest way to build data apps in Python. *Available at: https://streamlit.io/.*

[15] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* MIT Press.