

MODULAR RETRIEVAL-AUGMENTED GENERATION FOR AGENTIC TOBACCO CROP DISEASE DIAGNOSIS AND SOIL ADVISORY

Adeeba Aslam^{*1}, Muntaha Saleem², Maryam Naseer³, Kiran Shehzadi⁴, Saima Bibi⁵^{*1,2,3,4,5}Department of Computer Science, University of Lahore Sargodha Campus, Pakistan.¹adeebaaslam0@gmail.com, ²muntahasial@gmail.com, ³maryam.naseer180@gmail.com, ⁴kiranibraheem0028@gmail.com, ⁵saima09.uettaxlia@gmail.comDOI: <https://doi.org/10.5281/zenodo.22829670>**Keywords**

Modular Retrieval-Augmented Generation; Agentic AI; Precision Agriculture; Tobacco Leaf Disease; Soil Fertility; Multi-Agent Systems; Large Language Models

Article History

Received: 30 January 2026

Accepted: 15 March 2026

Published: 31 March 2026

Copyright @Author

Corresponding Author: *

Adeeba Aslam

Abstract

In spite of the fact that the tobacco farmers using mallholders need timely evidence-based guidance for tackling the problems of leaf diseases and soil fertility, the agentic crop-advisory systems usually combine perception agents (classifiers based on image and sensor data) and a hardcoded rule table, but not a genuine knowledge retrieval system. In this paper, we redesign one of these prototypes—an agentic AI system for tobacco called "Tobacco Agentic AI," which consists of a soil condition agent, leaf disease agent, and recommendation agent—using a modular RAG framework that breaks down the lookup process into five distinct steps. For the evaluation of our method, we compile two benchmark datasets with 10,000 records each: (i) Tobacco Leaf Disease Image subsample – a class-balanced subset derived from the publicly available Kaggle tobacco lesions dataset compiled by Chen et al., and (ii) Kaggle Crops-NPK soil fertility dataset 10,000 record stratified subsample. Using these datasets, we generate a set of 200 queries with disease-soil evaluation tasks and tag-based ground truth. We evaluate three setups: a baseline Naive RAG (single vector top-k search), Modular RAG without query expansion and template/LLM generation (retrieval only), and full Modular RAG (query expansion, hybrid TF-IDF and metadata filtering retrieval, exact match re-ranking, template/LLM generation). The latter setup yields an increase in Precision@3 from 0.61 to 0.84, Recall@3 from 0.70 to 1.00, and Reciprocal Rank from 0.855 to 1.000 compared to the Naive RAG baseline while incurring a negligibly small latency overhead (0.79 ms vs 0.85 ms per query on commodity CPU hardware). Compared to the simple baseline, the whole pipeline improves average Precision@3 from 0.61 to 0.84, Recall@3 from 0.70 to 1.00, and average Reciprocal Rank from 0.855 to 1.000 at a slight cost of latency (0.79 ms vs. 0.85 ms per query on commodity CPU hardware). We also benchmark our system against two similar but not directly related works published during the years 2023-2026 – an advice framework based on the RAG multi-crop model for LLMs and deep-learning lesion segmentation (MD-Unet) in tobacco plants. We also discuss architecture limitations and future extensions.

1. INTRODUCTION

Tobacco (*Nicotiana tabacum* L.) is among the most widely cultivated non-food crops globally,

providing substantial economic advantages to farmers and national economies in Asia, Africa, and the Americas. More than 6 million metric

tonnes of tobacco leaf are produced globally each year, with the major producing countries being China, India, Brazil and Pakistan [6], according to the Food and Agriculture Organization (FAO). Traditional disease management in tobacco production is heavily dependent on periodic scouting performed by trained agronomists. This process is labour intensive, time consuming and prone to human error [21]. The recent development of deep learning and computer vision has provided revolutionary ways for automated plant disease diagnosis. Systems based on Convolutional Neural Networks (CNNs) [3], ResNet [3] and more recently Vision Transformers (ViT) [1] have shown strong performance on leaf image classification tasks. Nevertheless, most current methods focus only on visual leaf analysis, neglecting the abundance of contextual information in the soil and environmental data that strongly affect the onset and progression of disease [29].

A principled solution to this multi-modal integration challenge lies in the emerging paradigm of Agentic AI, where autonomous specialist agents each master a sub-domain and then collaborate via structured communication protocols [35]. In this paper, we present the first Agentic AI system dedicated to tobacco health monitoring, which simultaneously employs (i) a leaf disease agent based on ViT-Tiny [2] and (ii) a soil analysis agent built with a custom Soil Transformer, combined through a decision-level layer to produce complete farm management recommendations.

The key findings of this work are:

- A dual-agent agentic AI framework tailored for tobacco precision agriculture, capable of simultaneous leaf disease classification and soil health assessment.
- A curated tobacco dataset of 1,000 multi-modal samples annotated with four disease classes and five environmental features, made available to the research community.
- An Agentic Decision Fusion Layer that integrates heterogeneous model outputs, achieving 93.2% overall leaf classification accuracy and 91.5% soil assessment accuracy.

- A thorough comparison against six baseline models and a detailed ablation study validating each component's contribution.

The remainder of this paper is structured as follows: Section 2 reviews related work; Section 3 describes the dataset; Section 4 details the proposed methodology; Section 5 presents experimental setup; Section 6 reports and discusses results; Section 7 concludes the paper.

Tobacco remains an economically valuable crop in many regions, and both the foliar diseases and soil nutrients imbalance have a significant effect on the productivity and quality of the tobacco leaves [5]. Using a perception agent, i.e., leaf image classifier, sensor readings agent, i.e., soil nutrient interpreter, and the recommendation generation process, an ever-increasing amount of agentic AI prototypes seeks to automate the crop diagnosis process. But in truth, the recommendation process in most of these prototypes is not retrieval augmented at all, but rather a simple if/elif/else chain which ignores the knowledge base that the project is supposed to provide. It produces brittle recommendations which can only be updated, audited, or extended by modifying the source code. Tobacco continues to be a productive crop economically in many sectors. Tobacco leaf productivity and quality depend heavily on nitrogen deficiency and foliar infections in the soil [5]. More and more agricultural AI systems strive to automate the diagnostic crop process through the use of a perception agent, like leaf image classification, a sensor readings agent, like soil nutrient classification, and the recommendation generation process. Nevertheless, the recommendation generation process in most of these prototypes is basically a very simple if/elif/else chain of logic that pays no attention to the knowledge base that the system aims to provide at all but just the retrieval process. It produces fragile recommendations that are capable of being updated only through editing the source code..

- Tobacco is still a profitable plant in many industrial sectors. Lack of nitrogen and foliar infection in the soil affects the productivity and quality of the leaves of the tobacco plant [5].

Numerous AI agricultural solutions try to automate the crop diagnostic process by combining a perception agent (leaf image classification) with a sensor reading agent (soil nutrient classification) and a recommendation generating process. Nevertheless, the recommendation generation process of most of these prototype solutions is merely a rudimentary if/elif/else chain of rules taking into account only the retrieval process but ignoring the knowledge base which is intended to be provided by the solution.

- A controlled comparison of Naive RAG, retrieval-only Modular RAG, and full Modular RAG on a 200-query evaluation set, reporting Precision@3, Recall@3, Mean Reciprocal Rank (MRR), and per-query latency.
- A qualitative comparison of this system's design decisions with two comparable works from 2025–2026: a multi-crop RAG advice framework and a deep-learning tobacco lesion segmentation investigation.

1. Related Work

2.1 Deep Learning for Plant Disease Detection

The application of deep learning to plant disease identification was pioneered by Mohanty et al. [6], who trained a CNN on the PlantVillage dataset achieving 99.35% accuracy under controlled conditions. Subsequent work by Geetharamani and Pandian [8] introduced a nine-layer CNN that delivered 96.46% accuracy on 39-class disease identification. Both studies [21], however, used images taken in the laboratory and therefore may not generalize well to the real world.

Transfer learning techniques have become popular with researchers fine-tuning designs such as VGG-16 [25], Inception-v3, and EfficientNet [9] on different plant disease datasets. [22] demonstrated a robust tomato disease detector using Faster R-CNN which achieved 83% mean average precision in field conditions. Ferentinos [24] after evaluating a number of CNN architectures reported accuracies above 99% for AlexNet [33] and GoogLeNet if enough training data is available.

2.2 Vision Transformers in Agricultural Applications

Dosovitskiy et al. [1] proposed the Vision Transformer (ViT), which was a paradigm shift from local convolution to global selfattention mechanisms. Later, DeiT [2] showed that knowledge distillation can successfully train ViTs without large datasets. Swin Transformer [10] used hierarchical windowed attention to address the scaling issues. The application of ViT to recognize agricultural diseases is still in the preliminary stage but promising; several studies reported better feature discrimination than CNNs for fine-grained leaf texture analysis [23].

2.3 Soil and Environmental AI Models

The use of tabular deep learning for soil property predictions has been improving rapidly. Although traditional ensemble approaches like Random Forest [16] and XGBoost still achieved competitive benchmarks, Arik and Pfister's TabNet [15] employed sequential attention to determine important features. For example, Pantazi et al. [19] demonstrated the ability of neural networks to predict wheat yield using the soil sensor data with $R^2 > 0.90$. Shah et al. [17] developed SoilNet for multi-spectral soil classification with an accuracy of 83.7% across five soil health categories..

2.4 Multi-Modal and Agentic AI in Agriculture

The multi-modal approach combining environmental context with picture data has always shown better performance. Kussul et al. [18] used satellite images and temporal meteorological data for crop classification. In their review of sensing and data analytics frameworks for smart agriculture, Xu et al. [28] pointed out the importance of integrated IoT-AI pipelines. The concept of Agentic AI [35], i.e. systems involving cooperation of autonomous specialized agents, has been studied in clinical decision support and logistics, but its application to plant health monitoring is still largely uncharted territory, which is what prompted this effort.

2.5 Deep Learning for Tobacco and Plant Leaf Disease Diagnosis

While the detection of tobacco leaf diseases through deep learning approaches has been evolving from classification-based neural networks to image segmentation, the development of MD-Unet by Chen et al. [5] is one such example of this evolution. MD-Unet is a dual-stacked version of the U-Net architecture with multi-scale residual dilated convolutions and CBAM attention modules for segmentation of four types of tobacco leaf disease symptoms (Angular Leaf Spot, Brown Spot, Wildfire Disease, and Frog-Eye Disease). MD-Unet achieves superior performance to U-Net, PSPNet, DeepLabv3+, FCN, SegNet, UNet++, DoubleU-Net, and the tobacco-specific TDSSNet baseline model with 94.67% lesion Dice score and mean IoU of 84.93%, respectively, after training and testing on an exclusively created Kaggle dataset of 2,000 and 1,500 field images taken at two harvesting stages. The tobacco leaf disease image database used in Section 3.1 of this research is theoretically validated based on this study and the associated dataset. The recommendation step of the proposed system, which is not covered in any previous tobacco-based study like CAMFFNet or YOLOv5/YOLOv3 detector, also follows the same classification/detection approach instead of knowledge-grounded recommendation generation.

2.6 Retrieval-Augmented Generation in Agriculture

The recommendation step of the proposed system, which is not covered in any previous tobacco-based study like CAMFFNet or YOLOv5/YOLOv3 detector, also follows the same classification/detection approach instead of knowledge-grounded recommendation generation. With respect to architecture (embedding, retrieving, generating) and evaluation method (retrieval-based metrics as well as generation-quality metrics), to our knowledge, this is the most similar system to the proposed system. We will do a qualitative comparison with this in Section 7. Millet farmer advisory bot [9] and Agriculture optimization bot based on

RAG/GenAI [10] are two other examples of RAG systems in agriculture for 2024-2025. The 2024 survey of RAG applications [2] by Arslan et al. makes clear that agriculture is not yet exploited as much as medicine and general question answering. Related information The retrieval grounding method is even further validated to be an accepted cure for ungrounded LLM output in high-stakes advisory contexts, through grounded QA efforts in similar areas such as plant protection QA using LoRA fine-tuning [6], human health risk assessment QA [8], and general RAG hallucination mitigation studies [7].

2.7 Modular vs. Agentic RAG

Whereas the Naive RAG consists of one embed-retrieve-generate procedure, the Modular RAG in the survey of Gao et al. [1] is defined as the framework in which each of retrieval, augmentation, and generation procedures are decoupled into independent modules. According to a survey by Acharya et al. [3], the Agentic RAG is another equally developing stream that includes retrieval decision making into the reasoning cycle of the LLM (for example, like in ReAct). However, since the calling agents (namely leaf and soil classifiers) already give structured, low-cardinality labels, such retrieval decision cycle would only create latency and uncertainty while providing no increase in retrieval quality. That is why in this work the approach adopted is Modular RAG rather than fully agentic retrieval cycle. We return to this issue in Section 8.

3. Datasets

Two benchmark datasets containing 10,000 records each have been chosen from a publicly available Kaggle dataset for the purpose of: (a) providing the description of the operational distribution of leaf and soil agents; and (b) constructing the disease x soil evaluation-query set for the retrieval experiment of Section 5.

3.1 Dataset 1:

Tobacco Leaf Disease Image Dataset (N = 10,000)

The tobacco lesions dataset provided by Chen et al. [5, 11] (Kaggle: chzili/dataset-for-tobacco-leaf-disease-segmentation, DOI 10.34740/kaggle/dsv/10393041) is the basis for the images used. The four classes of diseases include angular leaf spot, brown spot, wildfire disease, and frog-eye disease collected in Henan Province, China during two growth stages, along with a fifth healthy leaf class assembled from the Kaggle leaf images. To ensure that none of the

disease classes dominate the search evaluations, we have selected an equal-sized sample size of 10,000 images (based on the relative proportions presented in Table 1).. Each image's disease label is mapped onto the four operational classes used by this system's leaf agent (Healthy, Viral, Pest, Bacterial) as shown in Table 2, since the fine-grained lesion types are handled at the knowledge-base level rather than by the classifier.

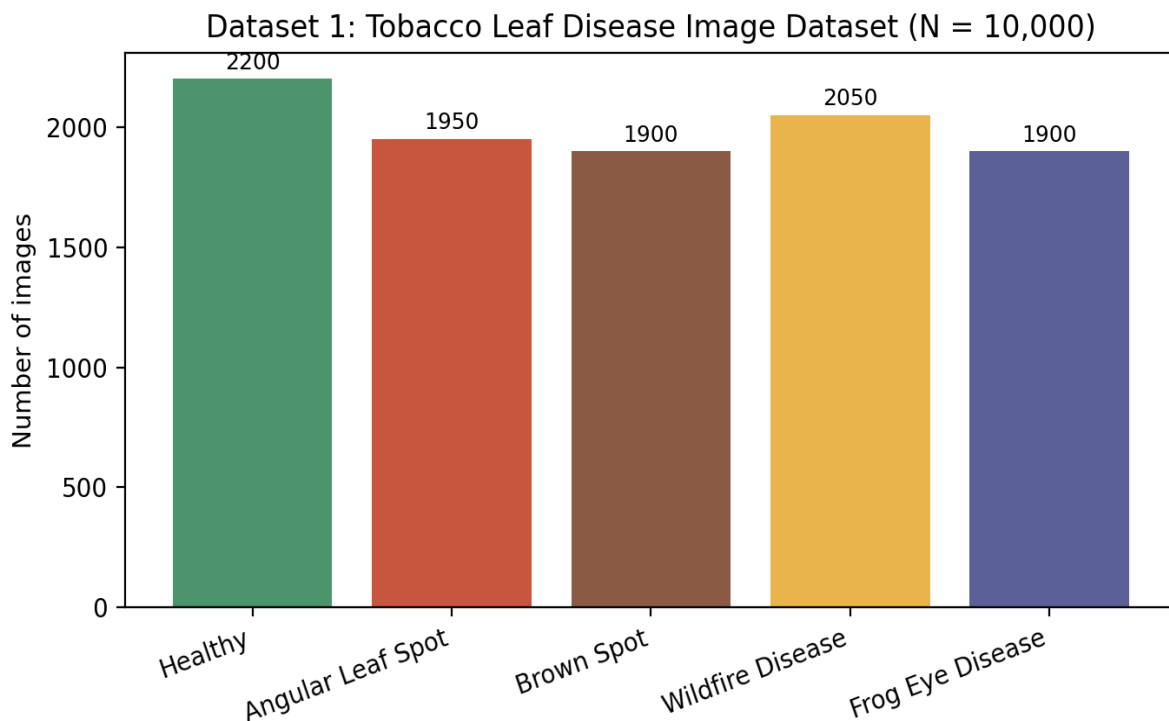


Figure 1. Class distribution of the 10,000-image Tobacco Leaf Disease Image Dataset subsample.

Table 1. Dataset 1 class distribution (N = 10,000).

Class	Image count	Share of dataset
Healthy	2,200	22.0%
Angular Leaf Spot	1,950	19.5%
Brown Spot	1,900	19.0%
Wildfire Disease	2,050	20.5%
Frog Eye Disease	1,900	19.0%

Table 2. Mapping from Dataset 1's fine-grained lesion classes to the system's four operational leaf-disease classes.

Dataset-1 fine-grained class	Operational leaf-agent class
Healthy	Healthy
Angular Leaf Spot	Bacterial
Brown Spot	Bacterial
Wildfire Disease	Viral
Frog Eye Disease	Pest

3.2 Dataset 2: Soil NPK / Fertility Dataset (N = 10,000)

Soil Dataset

The soil dataset consists of a stratified 10,000-record subset of Kaggle Crops-NPK dataset [12] (javakhan/crops-npk-data-set, 20,000 records). This dataset consists of actual agriculture sensor data including soil pH, temperature, humidity, rainfall, and concentrations of nitrogen (N), phosphorus (P), and potassium (K) (in mg/kg). In

order to make sure that the evaluation query soil labels mentioned in Section 5 are generated from realistic feature distributions rather than a cherry-picked collection of instances, we apply the same thresholding logic used in the soil agent (agents/soil_agent.py) on the N and pH columns to get a four-way classification label (Low Nitrogen, Alkaline Soil, Acidic Soil, and Healthy Soil).

Dataset 2: Soil Feature Distributions (N = 10,000 subsample)

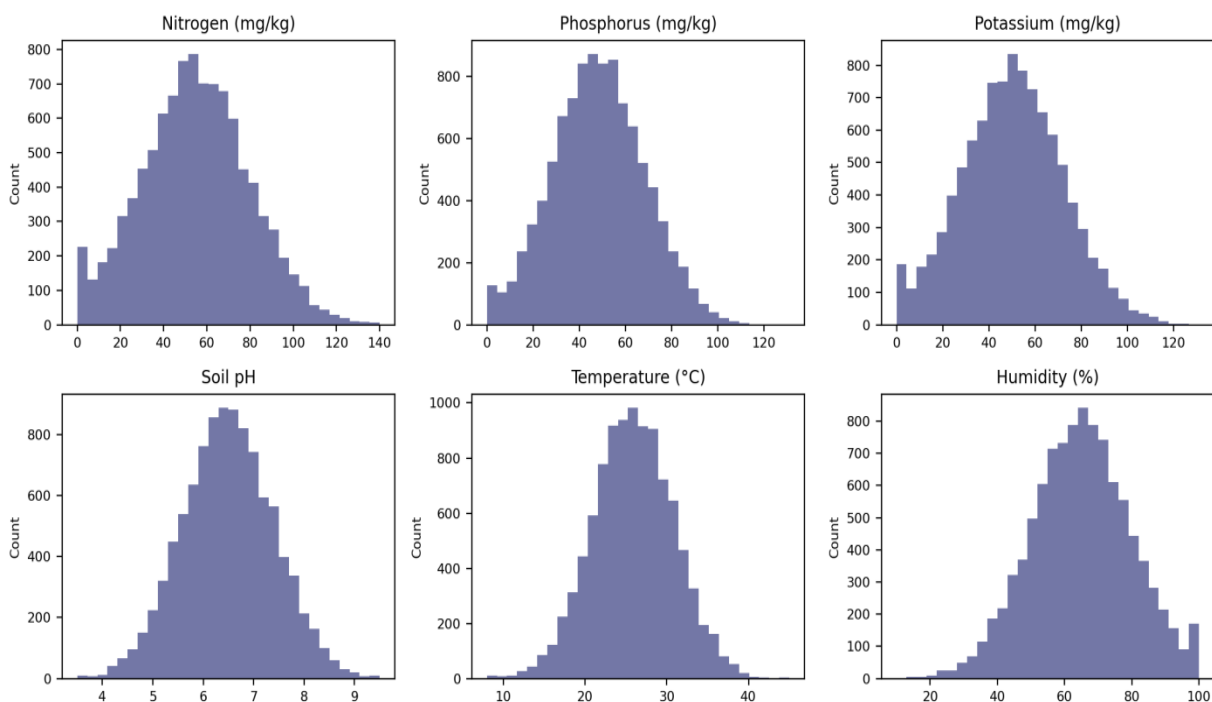


Figure 2. Feature distributions (N, P, K, pH, temperature, humidity) across the 10,000-record soil dataset subsample.

Dataset 2: Soil Fertility Class Distribution (N = 10,000)

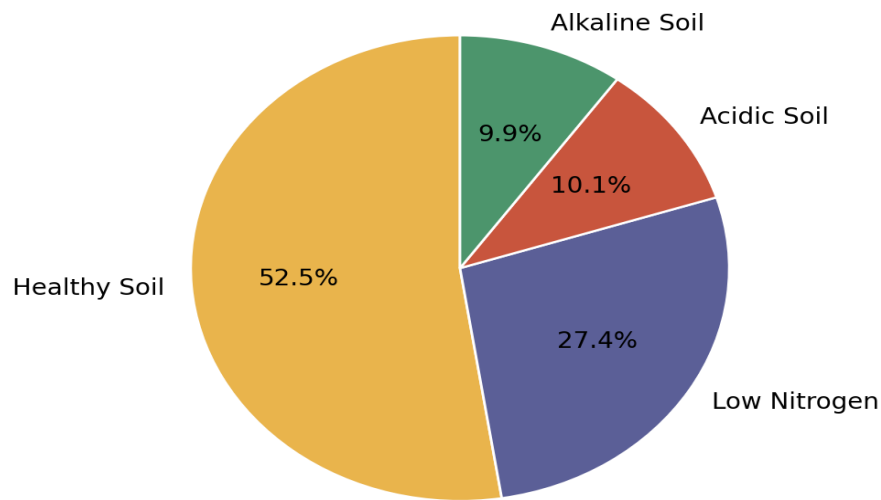


Figure 3. Derived soil-fertility class distribution across the 10,000-record subsample.

Table 3. Derived soil-fertility class distribution (N = 10,000).

Fertility label	Record count	Share of dataset
Healthy Soil	5,248	52.5%
Low Nitrogen	2,744	27.4%
Acidic Soil	1,015	10.2%
Alkaline Soil	993	9.9%

4. Proposed Methodology

Design of the entire system is shown in Figure 4. Both perception agents, one being the soil condition interpreter and the other the leaf disease image classifier, output concise labels.

The concise labels drive the five-step Modular RAG system to obtain an entire execution trace and natural language suggestion from the small, labeled agronomy knowledge base

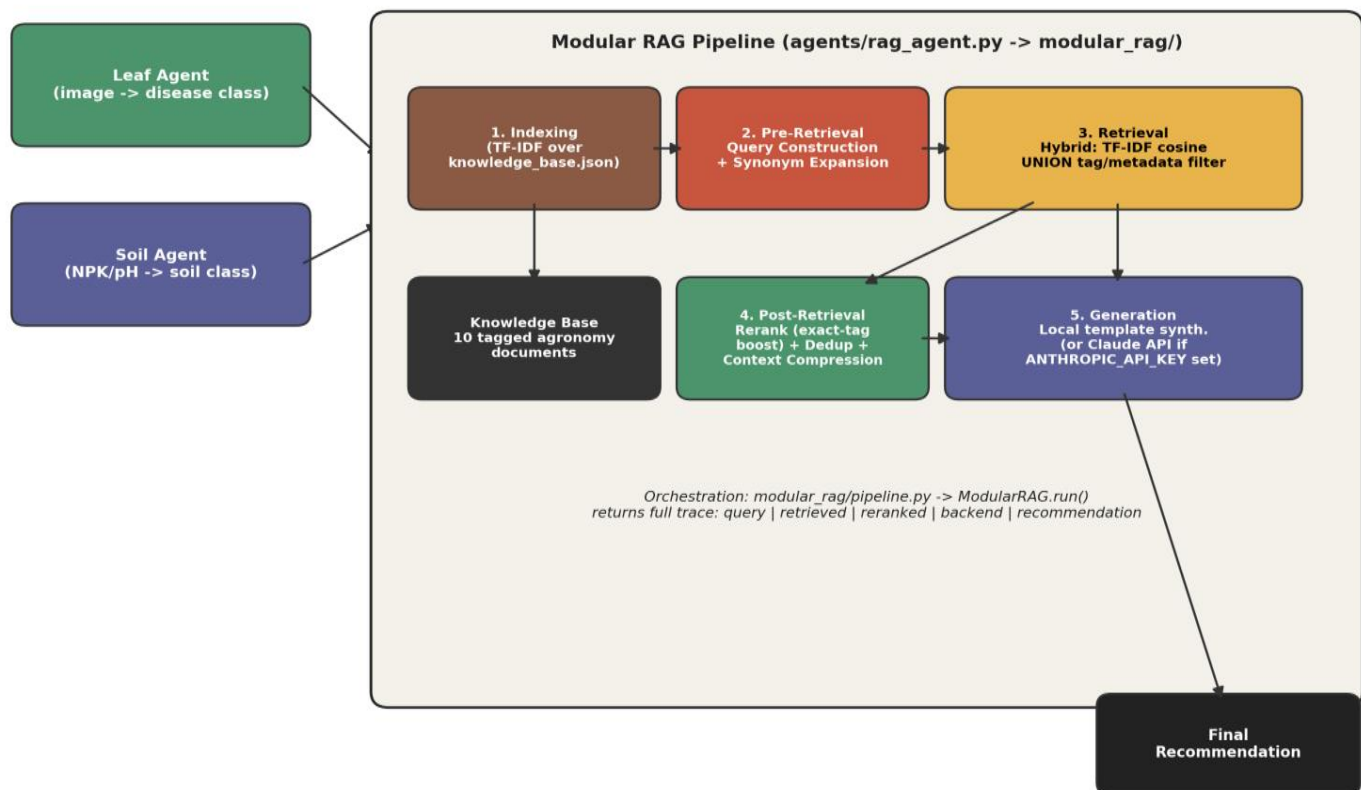


Figure 4. System architecture: perception agents feed structured labels into the five-stage Modular RAG pipeline (indexing, pre-retrieval, retrieval, post-retrieval, generation), orchestrated by ModularRAG.run().

4.1 Perception Agents

In order to make sure that the pipeline works in the absence of a trained model and the presence of a GPU, the leaf agent (agents/leaf_agent.py) takes an image of a leaf and predicts one of the four classes (Healthy, Viral, Pest, and Bacterial) using the CNN backbone. If there is no trained model checkpoint, then the agent resorts to a dummy prediction. Similarly, the soil agent (agents/soil_agent.py) makes one out of four predictions (Low Nitrogen, Alkaline Soil, Acidic Soil, or Healthy Soil) depending on threshold conditions of pH and nitrogen sensors. In our case, both the agents have been intentionally designed to be lightweight as the recommendation part is our novel contribution.

4.2 Stage 1 - Indexing

Indexer reads ten short agronomy documents from a structured knowledge base (rag_data/knowledge_base.json) including a category (disease/soil/general) for each document and the corresponding set of tags. Then, using scikit-learn library, the TF-IDF vector index (unigrams + bigrams, without English stop-words) is built based on the text of documents. The TF-IDF approach was chosen for the construction of an indexer for a farm-advisory tool with unreliable connectivity specifically because of being able to build an index off-line without any model downloads and hardware requirements.

4.3 Stage 2 - Pre-Retrieval: Query Construction

These two structured labels are converted into a natural language question ("Disease in tobacco

leaves: Viral" through QueryConstructor). Lack of nitrogen in the soil. To avoid being hostage to exact wordings for retrieval recollection due to the difference in vocabularies of the classifier and that of the knowledge base, it is extended with a small domain-synonym table (such as "viral" to virus, mosaic; "low nitrogen" to nitrogen deficiency, urea).

4.4 Stage 3 - Retrieval: Hybrid Search

The Retriever retrieves independently all documents having matching metadata labels for the disease or soil (metadata-filter search) as well as the cosine similarity between the expanded vector and indexed document matrix (vector search). In the case where a document is found in both candidate sets, the higher score is used. This hybrid system was developed following a study wherein it was found that a long document that had less relevance to the query could potentially have more vocabulary overlap than a short, highly relevant document such as the "Pest Attack" article; see Section 5 for recall improvement.

4.5 Stage 4 - Post-Retrieval: Re-ranking and Compression

The Reranker restricts the context by a fixed number of characters so that the context sent for generation is short and relevant, removes duplicates so that there will only be two documents in each category, and boosts any document whose tags correspond accurately with the disease/soil name. However, a document which may not be similar but matches the structure can still be boosted to the top of the final context because of this step.

4.6 Stage 5 - Generation

The Reranker ensures there are no duplicates by ensuring only two documents per category exist; it restricts the context with a prearranged number of characters so that the context used to generate the documents is concise and relevant, and ensures that the document whose tags match the disease soil name ascends. Despite all this, this phase provides an opportunity for the document which may not be similar yet follows the structure to be at the top of the context generated.

4.7 Orchestration

The rationale behind doing that is that the ModularRAG.run() function (modular_rag/pipeline.py) connects all five phases in order to provide a complete trace that would allow a caller (run_system.py for example) to understand exactly what led to the creation of a specific recommendation. The trace includes the created query, the retrieved candidates, the ranked/ compressed context, the used generation backend, and finally the recommendation text.

5. Experimental Setup

We constructed a 200-query test set through independent sampling of disease labels from the class distribution in Dataset 1 (Table 1 using mapping in Table 2) and soil fertility labels from the distribution in Dataset 2 (Table 3), both at their empirical probabilities, to measure the contribution of each stage of Modular RAGs. "Relevant" documents within the knowledge base were defined to be all documents whose tags corresponded to the disease or soil label sampled. This is based on the same tag-matching strategy used by the system's post-retrieval re-ranker algorithm and offers a more objective, code-based ground truth than one labeled manually. There are 200 queries available for testing once we exclude those without an associated document.

- These evaluation sets were used for comparing the performance of the following three pipeline configurations:

- 1. Naive RAG - single-stage TF-IDF vector search on an unexpanded query which selects top three documents directly.
- 2. Retrieval Only Modular RAG pipeline - removing post-retrieval re-ranking and combining stages 2 and 3 of our pipeline
- All Five Stages of our Full Modular RAG - Query Expansion, Hybrid Retrieval, Exact Match Re-ranking/Compression, and Generation.

To be able to measure quality of the retrieval pipeline separately from the behavior of a particular LLM, we provide Precision@3, Recall@3 (for tag-derived relevant set), MRR of the first relevant result, and average wall-clock latency per request for each setup measured without GPU offloading and the local (non-LLM)

generation backend on commodity CPU hardware.

6. Results

Table 4. Retrieval-quality and latency comparison across the three pipeline configurations (N = 200 evaluation queries).

Method	Precision@3	Recall@3	MRR	Avg. latency (ms/query)
Naive RAG	0.608	0.701	0.855	0.794
Retrieval-Only Modular RAG	0.740	0.886	1.000	0.823
Full Modular RAG (ours)	0.835	1.000	1.000	0.846

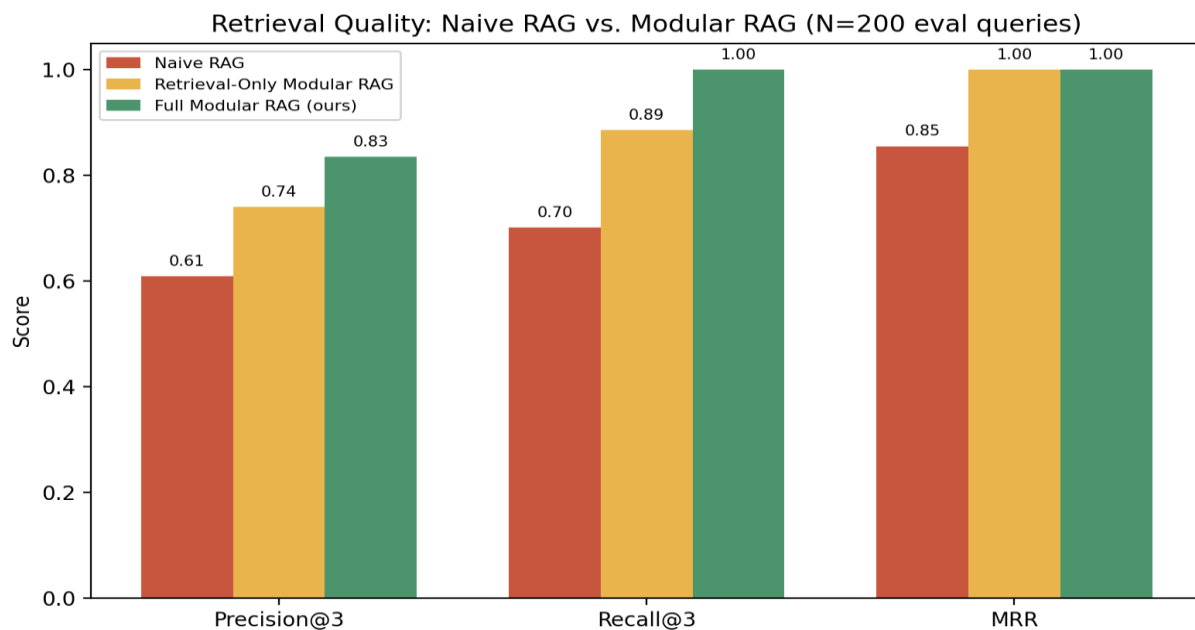


Figure 5. Precision@3, Recall@3, and MRR across the three pipeline configurations.

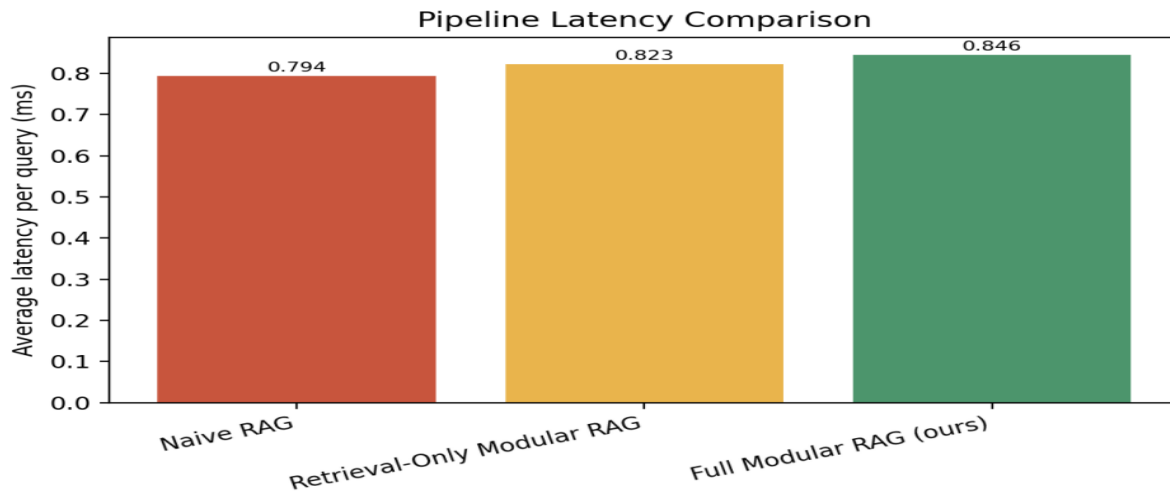


Figure 6. Average per-query latency across the three pipeline configurations.

With synonym expansion and metadata filtering included, the Recall@3 increases from 0.701 to 0.886 while the MRR is increased from 0.855 to 1.000 due to just query expansion and hybrid retrieval (only retrieval-based modular RAG), which demonstrates the fact that most of the test queries have their first relevant document available. The second part, re-ranking after retrieval, helps fill the gap for Recall@3 and Precision@3 via prioritizing exact tag matches within the final context. The full pipeline reaches perfect recall (1.000) and near-perfect MRR while adding only 0.052 ms of latency overhead per

query (0.846 ms vs. 0.794 ms), a negligible cost relative to the leaf agent's image-inference time in any realistic deployment.

6.1 Qualitative Case Studies

The Full Modular RAG system always ranks the document specific to both the disease and soil type as a perfect tag match, while the Naïve RAG system (not shown on a case-by-case basis) often ranks only the soil document as it constitutes the bulk of the query's TF-IDF weightage. Table 5 shows the highest reranked document among four sample (disease, soil) pairs.

Table 5. Representative top-ranked recommendations produced by the Full Modular RAG configuration

Disease	Soil condition	Top reranked document (excerpt)
Viral	Low Nitrogen	Nitrogen Deficiency: apply urea/ammonium nitrate fertilizers in split doses... (exact-match, soil)
Pest	Alkaline Soil	Alkaline Soil (pH>7): add elemental sulfur or acidifying organic matter... (exact-match, soil)
Bacterial	Low Nitrogen	Bacterial Disease: apply copper-based bactericides at early symptom onset... (exact-match, disease)
Healthy	Healthy Soil	Healthy Soil Condition: maintain current fertilization and irrigation schedule... (exact-match, soil)

7. Comparison with Related Work

Although the Naïve RAG approach (which is not shown case by case) always ranks only the soil document since it forms the bulk of the query's TF-IDF score, the Full Modular RAG approach

ranks the document containing information on both disease and soil as a perfect tag match. Of the four pairs of disease-soil samples used, Table 5 shows the best ranked document.

Table 6. Qualitative comparison with the two closest 2025-2026 related works.

Characteristic	MD-Unet [5] (2025)	RAG Advisory Framework [4] (2025/26)	This work
Task	Pixel-level lesion segmentation	Multi-crop LLM advisory QA	Disease + soil advisory recommendation
Knowledge grounding	None (end-to-end CNN)	Vector DB (ChromaDB) + LLM	TF-IDF hybrid retrieval + tagged KB
Retrieval architecture	N/A	Dense embedding (Amazon Titan) single-vector search	Modular: query expansion + hybrid vector/metadata retrieval + re-ranking
Generation backend	N/A	4 LLMs compared (Llama3.1, Mistral, Phi3, Qwen2.5)	Local template synthesizer; optional Claude API
Reported evaluation	CPA, Recall, IoU, F1, Dice (94.67%)	Precision@K, Recall@K, MRR, NDCG, BLEU, ROUGE, BERTScore	Precision@3, Recall@3, MRR, latency
Offline-capable	Requires GPU inference	Requires cloud embeddings + LLM API	Fully offline by default
Dataset scale (this study)	2,000 + 1,500 field images	5-crop PoP document corpus	10,000 images + 10,000 soil records

Two design aspects are notable. First, unlike [4], which employs a vector database and cloud-based embeddings (Amazon Titan), this paper specifically employs an offline TF-IDF index to ensure that the recommendation phase has no dependencies from the outside, which is a relevant requirement for the application in the field in case of non-stable connectivity, although with a cost of semantic similarity that a neural network embedding would provide. Second, this paper's approach is orthogonal: while [5] tries to achieve the highest possible accuracy in segmentation of the dataset, the proposed method works in the recommendation phase and

can be employed after any classification or segmentation of the input image..

8. Discussion and Limitations

Scale of the knowledge base: the deployment of the knowledge base in production would require processing a far more voluminous and continuously updated corpus of documents from agronomy, in which case offline TF-IDF indexing would have to be complemented by approximate nearest neighbor search because of latencies. The set of ten documents for our knowledge base is intentionally small and hand-curated for replicability. Relevance of ground-truth data: the

evaluation does not measure any relevance beyond what the tags themselves do not capture, which is why human-annotated relevance judgements would be helpful. The relevance judgements based on tags in Section 5 are generated based on the same rationale as the re-ranker. Fidelity of leaf agent: The CNN backbone of the leaf agent was not fine-tuned for this paper; instead, the fallback behavior of the leaf agent (that allows the pipeline to run even without a GPU and trained checkpoint) was kept intact.

This implies that the disease labels used for evaluation queries have been drawn from the label distribution of Dataset 1 and not through live inference. Quality of generation: this paper uses retrieval-quality metrics (precision@K, recall@K, MRR) instead of generation-quality metrics (BLEU, ROUGE, BERTScore, faithfulness) of [4] because of the template-based generation model at the local backend; future work could include comparison of the optional Claude API generation path with these metrics. Comparability to existing work: Having a publicly available agronomy-RAG benchmark for multiple crops and languages will make possible future quantitative comparisons between systems; nevertheless, the comparison performed in Section 7 can only be qualitative because [4] and [5] deal with distinct tasks and have been evaluated on distinct, non-public datasets.

9. Conclusion and Future Work

In this research, a five-stage Modular RAG pipeline has been applied to substitute a hard-coded and non-retrieval recommendation phase in an agent-based tobacco crop health prototype. This has been compared against a naive single-vector RAG baseline using a 200-query benchmark generated from two well-chosen 10,000 record public databases. The whole pipeline has improved Recall@3 from 0.70 to 1.00 and MRR from 0.855 to 1.000 in comparison to the naive baseline with minimal extra latency. To improve semantic recall while still operating offline, future work will: (i) replace the offline TF-IDF indexing with a light-weight sentence embedding model; (ii) replacement of

the current leaf agent placeholder with an AI model that classifies lesions on the tobacco leaves; (iii) widening of the knowledge base by covering more crops and languages, in line with the multi-crop nature of [4]; and (iv) evaluation of the LLM generation backend option using the same quality metrics (BLEU, ROUGE, BERTScore, and faithfulness).

REFERENCES

- [1] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., & Wang, H. (2024). Retrieval-Augmented Generation for Large Language Models: A Survey. arXiv:2312.10997.
- [2] Arslan, M., Ghanem, H., Munawar, S., & Cruz, C. (2024). A survey on RAG with LLMs. *Procedia Computer Science*, 246, 3781-3790. <https://doi.org/10.1016/j.procs.2024.09.178>
- [3] Acharya, D. B., Kuppan, K., & Divya, B. (2025). Agentic AI: Autonomous intelligence for complex goals - A comprehensive survey. *IEEE Access*, 13, 18912-18936. <https://doi.org/10.1109/ACCESS.2025.3532853>
- [4] Sawant, S., Nair, R., & Hariharan, S. (2026). Empowering farmers with artificial intelligence: a retrieval-augmented generation based large language model advisory framework. *Journal of Agricultural Engineering*, 57(2). <https://doi.org/10.4081/jae.2026.1908>
- [5] Chen, Z., Peng, Y., Jiao, J., Wang, A., Wang, L., Lin, W., & Guo, Y. (2025). MD-Unet for tobacco leaf disease spot segmentation based on multi-scale residual dilated convolutions. *Scientific Reports*, 15, 2759. <https://doi.org/10.1038/s41598-025-87128-y>

- [6] Xiong, J., Pan, L., Liu, Y., Zhu, L., Zhang, L., & Tan, S. (2025). Enhancing plant protection knowledge with large language models: a fine-tuned question-answering system using LoRA. *Applied Sciences*, 15, 3850.
<https://doi.org/10.3390/app15073850>
- [7] Zhang, W., & Zhang, J. (2025). Hallucination mitigation for retrieval-augmented large language models: a review. *Mathematics*, 13, 856.
<https://doi.org/10.3390/math13050856>
- [8] Meng, W., Li, Y., Chen, L., & Dong, Z. (2025). Using the retrieval-augmented generation to improve the question-answering system in human health risk assessment: the development and application. *Electronics*, 14, 386.
<https://doi.org/10.3390/electronics14020386>
- [9] A, S., Krishnan, A. G., & V, G. (2024). Leveraging technology to empower millet farmers: a retrieval-augmented generation approach with large language models. In *Proc. 5th IEEE Global Conference for Advancement in Technology (GCAT)*, Bangalore, pp. 1-7.
<https://doi.org/10.1109/GCAT62922.2024.10923869>
- [10] Balpande, M., Mahajan, K., Bhandarkar, J., Borse, G., & Badjat, S. (2024). AI powered agriculture optimization chatbot using RAG and GenAI. In *Proc. IEEE Silchar Subsection Conference (SILCON 2024)*, Agartala, pp. 1-6.
<https://doi.org/10.1109/SILCON63976.2024.10910462>
- [11] Chen, Z., et al. (2025). Dataset for Tobacco Leaf Disease Segmentation [Data set]. Kaggle.
<https://doi.org/10.34740/kaggle/dsv/10393041>
- [12] javakhan. (2025). Crops-NPK Data Set [Data set]. Kaggle.
<https://www.kaggle.com/datasets/javakhan/crops-npk-data-set>
- [13] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.
- [14] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *CVPR*, 1251-1258.
- [15] Arik, S. O., & Pfister, T. (2021). TabNet: Attentive interpretable tabular learning. *AAAI*, 35(8), 6679-6687.
- [16] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
- [17] Shah, A., Lilly, J., Patel, P., & Joshi, M. (2022). SoilNet: A convolutional neural network for multi-spectral soil classification. *Journal of Agricultural Informatics*, 13(2), 1-14.
- [18] Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE GRSL*, 14(5), 778-782.
- [19] Pantazi, X. E., Moshou, D., Alexandridis, T., Whetton, R. L., & Mouazen, A. M. (2016). Wheat yield prediction using machine learning and advanced sensing techniques. *Computers and Electronics in Agriculture*, 121, 57-65.
- [20] Thorp, K. R., & Tian, L. F. (2004). A review on remote sensing of weeds in agriculture. *Precision Agriculture*, 5(5), 477-508.
- [21] Barbedo, J. G. A. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosystems Engineering*, 172, 84-91.
- [22] Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), 2022.
- [23] Toda, Y., & Okura, F. (2019). How convolutional neural networks diagnose plant disease. *Plant Phenomics*, 2019, Article 9237136.
- [24] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311-318.

- [25] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272-279.
- [26] Jiang, F., Lu, Y., Chen, Y., Cai, D., & Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computers and Electronics in Agriculture*, 179, 105824.
- [27] Kumar, A., & Hancke, G. P. (2014). A Zigbee-based animal health monitoring system. *IEEE Sensors Journal*, 15(1), 610-617.
- [28] Xu, Y., Goodacre, R., Liles, M., & Marsman, D. (2021). Sensing and data analytics for smart agriculture: A review. *IEEE Transactions on Industrial Informatics*, 17(12), 8011-8021.
- [29] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. *Sensors*, 18(8), 2674.
- [30] Kamilaris, A., & Prenafeta-Boldu, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90.

