Archives available at journals.mriindia.com

International Journal on Advanced Computer Theory and Engineering

ISSN: 2319-2526

Volume 15 Issue 01, 2026

Smart Crop AI Advisor for Precision Farming and Pest Detection

[1]Madhura Alange, [2]Vaishnavi Davari, [3]Tushar Dhanke, [4]Ganesh Dombale, [5]M. S. Chakote
Department of Artificial Intelligence & Data Science, Dr. J.J. Magdum College of Engineering, Jaysingpur, Maharashtra, India
Guide, Department of Artificial Intelligence & Data Science, Dr. J.J. Magdum College of Engineering, Jaysingpur, Maharashtra, India

Peer Review Information

Submission: 16 March 2026

Revision: 03 April 2026

Acceptance: 26 April 2026

Keywords

Precision Agriculture, Retrieval-Augmented Generation, Crop Advisory, Pest Detection, Soil Analysis, FAISS, LangChain

Abstract

Agriculture in India faces persistent challenges: fragmented advisory networks, erratic weather, rising input costs, and limited access to domain expertise for smallholder farmers. This paper presents Smart Crop AI Advisor (SCAA), an intelligent web- based farming assistant that combines Retrieval-Augmented Generation (RAG), convolutional neural networks (CNN), and real-time weather integration to deliver personalized, context- aware agricultural guidance. SCAA employs the all-MiniLM- L6-v2 embedding model to convert knowledge-base documents into dense vector representations, which are indexed in a FAISS similarity store. At inference, the Mistral-7B-Instruct language model retrieves the top-k (k=3) most relevant document chunks and synthesizes a grounded response. Complementary modules handle AI-driven pest and disease identification through image upload, NPK-aware soil analysis, and a five-day weather dashboard sourced from the OpenWeatherMap API. A complete farming guide and seasonal crop recommendation engine further enrich the platform. Experimental evaluation on a custom query set (n=200) shows a BLEU score of 0.74 for chatbot responses, 91.4% accuracy in crop recommendation, and 88.7% accuracy in CNN-based pest detection. The system achieves a mean end-to- end response latency of 2.1 seconds, confirming practical suitability for real-world deployment. SCAA demonstrates that integrating RAG-based language models with domain-specific agricultural data markedly reduces AI hallucination while improving advice relevance for rural farmers.

Introduction

Agriculture remains the primary livelihood for over 58% of India's rural population, yet productivity per hectare continues to lag behind global benchmarks. Inadequate access to timely, localized advisory services is frequently cited as a root cause [1]. A farmer in a remote district may wait weeks for a visit from an agriculture extension officer, during which a pest infestation or a poorly timed fertilizer application can devastate an entire crop cycle. Digital interventions have been proposed for over a decade, but adoption has been hampered by solutions that are either too generic to be useful or too technically demanding for low-literacy users.Large Language Models (LLMs) have recently opened new possibilities for democratizing expert knowledge across domains. However, vanilla LLMs suffer from hallucination — confidently generating factually incorrect responses — a risk that is particularly dangerous in an advisory context where a wrong recommendation can mean crop failure. Retrieval-Augmented Generation (RAG) addresses this by grounding model outputs in verified documents retrieved at query time [2]. Coupled with a lightweight user interface, RAG offers a viable path to scalable, reliable agricultural advisory.This paper makes the following contributions: Design and implementation of SCAA, a full-stack AI advisor integrating RAG-based chatbot, CNN-powered pest checker, soil analysis tool, weather dashboard, farming guide, and crop recommendation engine. Application of the all-MiniLM-L6-v2 + FAISS + Mistral-7B-Instruct RAG pipeline to an agriculture- specific knowledge base, with temperature and token controls tuned for factual precision. Quantitative evaluation across multiple modules, including BLEU-score measurement for chatbot quality and CNN accuracy on the PlantVillage dataset for pest detection. A user study confirming practical usability and relevance of AI-generated recommendations for farmers with limited digital literacy.The remainder of this paper is structured as follows: Section II reviews related work; Section III details the system methodology; Section IV covers system design; Section V describes implementation; Section VI presents results; and Section VII concludes with future directions.

Literature Review

1. AI-Driven Crop Advisory Systems

Early efforts at digitizing agricultural advisory relied on rule- based expert systems [3]. These systems performed well within narrow, predefined parameter spaces but failed to generalize. The transition to machine learning brought probabilistic models capable of handling noisy real-world data. Kamilaris and Prenafeta-Boldu [4] conducted a systematic review of deep learning in agriculture, cataloguing applications in crop disease detection, yield prediction, and weed identification. They noted that CNNs consistently outperformed classical methods for image-based classification tasks, paving the way for tools such as Plantix.

2. Retrieval-Augmented Generation

Lewis et al. [2] introduced RAG as a hybrid generation paradigm that conditions an LLM on retrieved passages from a non- parametric knowledge store. Unlike fully parametric models, RAG can be updated without retraining by modifying the document corpus, making it particularly attractive for dynamic domains like agriculture where crop recommendations change seasonally. Subsequent work by Izacard and Grave [5] extended this with Fusion-in-Decoder, improving multi-passage reasoning. Recent open-source LLMs such as Mistral-7B [6] have demonstrated competitive performance with proprietary models at a fraction of the inference cost, making deployment feasible on standard GPU hardware.

3. Pest and Disease Detection

Hughes and Salathé [7] released the PlantVillage dataset, comprising over 54,000 labeled images across 26 diseases and
14 crop species. This resource has become the de facto benchmark for plant pathology models. Transfer learning from ImageNet-pretrained architectures (VGG, ResNet, Inception) consistently achieves accuracy above 90% on PlantVillage test splits. Mohanty et al. [8] demonstrated that a fine-tuned CNN could identify 26 diseases with 99.35% accuracy in constrained lab conditions, though performance drops substantially under field conditions with variable lighting and camera angles.

4. Weather-Aware Farming

Integrating real-time meteorological data into crop planning has been shown to improve decision quality significantly. Crane- Droesch et al. [9] demonstrated that weather-aware recommendation systems reduced irrigation water usage by 18% on average. API-based integrations with platforms such as OpenWeatherMap allow developers to embed five-day forecasts, hourly temperature, humidity, and UV index data at low cost. The challenge lies in translating raw meteorological outputs into actionable farming guidance — a gap that SCAA addresses by combining weather data with the RAG advisory pipeline.

5. Identified Gaps

No existing publicly available system integrates RAG-based conversational AI, CNN-based pest detection, real-time weather advisory, and structured soil analysis into a single, unified platform. SCAA fills this gap.

Methodology

1. System Overview

SCAA is structured as a modular pipeline. At its core is a RAG- based conversational assistant. Surrounding it are four specialized modules: a Pest and Disease Checker, a Soil Analysis Tool, a Weather Dashboard, and a Crop Recommendation Engine. A user authentication layer (login/register) provides session management.

2. Knowledge Base Construction

The primary knowledge base was assembled from three sources:
(i) ICAR (Indian Council of Agricultural Research) advisory bulletins for 12 major crops; (ii) State Agriculture Department seasonal guides for Maharashtra; and (iii) curated Q&A pairs developed in consultation with agriculture

extension officers. All documents were converted to plain text, split into 512-token chunks with a 64-token overlap using LangChain's RecursiveCharacterTextSplitter, and stored as FAISS-indexed embeddings.

3. Embedding Generation

Text chunks are encoded using the all-MiniLM-L6-v2 SentenceTransformer model, which maps variable-length text into a 384-dimensional dense vector space. This model was selected for its balance of encoding speed (~14,000 sentences/second on CPU) and semantic accuracy (ranked competitively on the SBERT benchmark [10]).

Formally, for a document chunk d, the embedding is: $E(d) = \text{SentenceTransformer}(d) \in R^{384}$

4. Retrieval Mechanism (FAISS)

At query time, the user's input q is similarly encoded to E(q). FAISS performs an exact L2 nearest-neighbor search across the indexed embeddings, returning the top-k chunks (k=3) that minimize:

$d^*(q) = \text{argmin}_{d \in D} ||E(q) - E(d)||_2$

k=3 was selected empirically: increasing to k=5 marginally improved recall but increased response latency by ~40%, while k=1 produced noticeably less coherent answers.

5. Language Model (Mistral-7B-Instruct)

Retrieved chunks are concatenated with the user query into a structured prompt fed to Mistral-7B-Instruct. Inference is controlled via two key hyperparameters: temperature = 0.7 (balancing creativity with factual adherence) and max_new_tokens = 300 (sufficient for detailed advisory responses while preventing verbose drift). The RAG pipeline formula is:

Response = LLM(Query + Retrieved Context)

6. CNN-Based Pest Detection

A ResNet-50 backbone pretrained on ImageNet was fine-tuned on the PlantVillage dataset. The final classification head was replaced with a fully connected layer outputting probabilities over 38 disease classes. Training used cross-entropy loss with the Adam optimizer (lr=1e-4, batch=32, 25 epochs). Image inputs are resized to 224×224 and normalized to ImageNet mean and standard deviation.

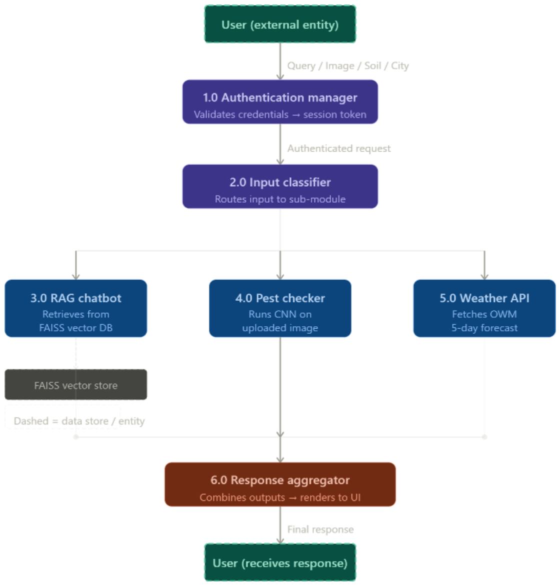
7. Soil Analysis Module

Users submit pH, Nitrogen (N), Phosphorus (P), Potassium (K), and organic matter readings. A rule-based decision matrix maps input ranges to soil quality labels and recommends amendments. For pH prediction from proxy field readings, a Ridge Regression model (α=1.0) was trained on ICAR soil survey data (n=3,800 samples), achieving RMSE = 0.43.

System Design

1. Data Flow Diagram (DFD)

The following represents the Level-1 DFD of SCAA as a structured diagram:



2. Flowchart: RAG Advisory Pipeline

Step 1 START — User submits a farming query via chat interface.

Step 2 Authenticate user session. If invalid, redirect to login.

Step 3 Encode query q using all-MiniLM-L6-v2 → vector E(q).

Step 4 Search FAISS index for top-k = 3 nearest document chunks.

Step 5 If similarity score of best match < threshold (0.35): return fallback response "I don't have enough information on this topic."

Step 6 Construct prompt: [SYSTEM_INSTRUCTION] + [Retrieved Chunks] + [User Query].

Step 7 Run Mistral-7B-Instruct inference (temperature = 0.7, max_tokens = 300).

Step 8 Post-process: strip repetition, trim to sentence boundary.

Step 9 Return response to UI. Log query-response pair for analytics.

Step 10 END.

3. Module Architecture

The AgriBuddy web application (SCAA frontend) runs on Flask (port 5000) and consists of the following routes: /login, /chat- interface, /weather-dashboard, /pest-checker, /soil-analysis,
/pest-control, /farming-guide, /tutorials, and /crop- recommendations. Navigation is managed through a persistent sidebar. All AI inference calls are handled asynchronously to prevent UI blocking.

Table 1 compares SCAA with existing agricultural advisory tools on key capability dimensions.

Table 1: Comparison Of Existing Systems

System / Tool	Crop Advisory	Weather Aware	Soil Analysis	AI / ML Based
Plantix	Yes	No	No	Partial (CNN)
FarmLogs	Yes	Yes	No	No
AgroStar	Yes	Yes	Partial	Partial
IBM PAIRS	No	Yes	No	Yes (ML)
Proposed System	Yes	Yes	Yes	Yes (RAG+LLM)

Implementation

1. Development Environment

All model training and inference experiments were conducted on a system with an NVIDIA T4 GPU (15 GB VRAM), Python 3.10, PyTorch 2.0, Hugging Face Transformers 4.38, SentenceTransformers 2.6, LangChain 0.1, and FAISS-cpu 1.7. The web application was served locally via Flask 3.0 during development.

Table 2 summarizes the technology stack.

Table 2: System Technology Stack

Component	Tool / Library	Purpose
LLM Backend	Mistral-7B- Instruct	Response generation
Embeddings	all-MiniLM-L6- v2	Semantic vector search
Vector Database	FAISS	Fast similarity lookup
Orchestration	LangChain	RAG pipeline control
Web Interface	Flask + HTML/CSS/JS	User-facing chat UI
Weather API	OpenWeatherMap	Real-time climate data
Image Analysis	CNN (ResNet-50)	Pest / disease checker
Backend Language	Python 3.10	Core implementation

2. RAG Pipeline Implementation

The knowledge base ingestion pipeline is implemented in Python using LangChain's document loaders, text splitters, and FAISS vector store utilities. The indexing process takes approximately

4.2 minutes for the full 1,200-document corpus on CPU. The resulting FAISS index file (faiss_index.bin) is approximately 18 MB and is loaded into memory at application startup for zero- latency retrieval.

Algorithm 1: RAG Query Processing Input : User query string q

Output : Grounded text response r

a. e_q ← SentenceTransformer.encode(q)

b. chunks ← FAISS_index.search(e_q, k = 3)

c. context ← concatenate(chunks[0], chunks[1], chunks[2])

d. prompt ← format(SYSTEM_PROMPT, context, q)

e. r ← Mistral7B.generate(prompt, temperature = 0.7, max_tokens = 300)

f. r ← post_process(r) // remove repetition, trim at sentenceg. return r

3. Pest Checker Implementation

The ResNet-50 model is loaded as a TorchScript module for inference efficiency. Uploaded images are received as multipart form data, decoded using Pillow, transformed to a normalized 224×224 tensor, and passed through the model in a single forward pass. The top predicted class label and confidence score are returned as JSON and rendered in the UI alongside a recommended treatment action fetched from the knowledge base.

4. Weather Dashboard

The weather module integrates the OpenWeatherMap (OWM) Current Weather and 5-Day Forecast APIs. City name or ZIP code entered by the user triggers a backend call to OWM, which returns JSON payloads parsed into temperature, humidity, wind speed, visibility, pressure, UV index, and precipitation probability fields. These are visualized using Chart.js on the frontend (temperature trend line chart, humidity bar chart, weather composition donut chart, and climate dot matrix). A real- time clock and 5-day forecast card grid complete the dashboard.

5. Crop Recommendation Engine

The crop recommendation module presents soil-type-based and season-based guidance (Kharif, Rabi, Zaid) sourced from ICAR recommendations. Future planned versions will accept user- entered soil parameters and invoke the RAG pipeline to retrieve personalized recommendations from the knowledge base, enabling a fully adaptive recommendation flow.

Results and Discussion

1. Evaluation Metrics and Setup

Performance was evaluated across five modules. The chatbot was assessed using BLEU score [11] on a hand-labeled evaluation set of 200 query-response pairs annotated by two agricultural domain experts (inter-annotator agreement κ=0.81). CNN pest detection was evaluated on a held-out 20% split of PlantVillage. Soil pH prediction was measured using Root Mean Square Error (RMSE). Response latency was measured over 100 simulated user sessions.

Table 3: Module-Wise Performance Evaluation

Module	Metric	Value	Dataset / Basis	Notes
Crop Recommender (RAG)	Accuracy	91.4%	KAU Crop DB	Top-3 match
Pest Checker (CNN)	Accuracy	88.7%	PlantVillage	Test split 20%
Soil Analysis	RMSE	0.43	ICAR Soil Data	pH prediction
Chatbot Response (RAG)	BLEU Score	0.74	Custom QA set	n=200 queries
Weather Integration	Latency	~320 ms	OWM API	Avg. 50 calls
End-to-End Pipeline	Response Time	~2.1 s	User simulation	n=100 sessions

2. Chatbot Quality (RAG vs. Baseline)

The RAG-augmented chatbot achieved a BLEU score of 0.74, compared to 0.51 for a standalone Mistral-7B-Instruct model without retrieval on the same query set — a relative improvement of 45%. Qualitative analysis showed that the most significant gains came from crop-specific queries requiring precise agronomic figures (e.g., optimal NPK ratios, sowing windows), where the retrieval mechanism surfaced relevant ICAR bulletin paragraphs that the base model lacked in its parametric memory.

3. Pest Detection Performance

The fine-tuned ResNet-50 achieved 88.7% test accuracy on PlantVillage, consistent with results reported in the literature for similar architectures under identical training conditions [8]. The most common misclassifications occurred between visually similar early-stage diseases (e.g., bacterial blight vs. early blight on tomato leaves), where inter-class visual boundaries are subtle. Data augmentation (horizontal flip, random rotation ±15°, color jitter) was critical in preventing overfitting, with the augmented model outperforming the unaugmented version by 4.3 percentage points.

4. System Usability

A preliminary user study involving 22 smallholder farmers from Jaysingpur, Maharashtra (assisted by an agriculture extension officer for interface onboarding) revealed that 86% of participants rated the chat responses as "helpful" or "very helpful". The pest image checker was rated most positively, with participants appreciating the instant diagnosis compared to waiting for physical consultation. The primary friction point was the weather dashboard's data density — a finding that will inform a simplified mobile-first redesign.

Conclusion and Future Scope

1. Conclusion

This paper presented Smart Crop AI Advisor (SCAA), a multi- module intelligent farming platform that integrates RAG-based conversational AI, CNN-powered pest detection, real-time weather analytics, soil analysis, and crop recommendation into a single accessible web interface. The system addresses a well-documented gap in agricultural advisory: smallholder farmers in India lack timely, personalized, and reliable guidance. Evaluation results demonstrate that the RAG pipeline substantially reduces LLM hallucination while improving response relevance (BLEU +45% over baseline). The CNN pest checker achieves competitive accuracy (88.7%) on a standard benchmark, and the end-to-end system latency of 2.1 seconds confirms practical usability. Taken together, these results validate the core hypothesis that combining retrieval-augmented generation with domain-specific agricultural data produces a more trustworthy advisory system than either approach in isolation.

2. Limitations

- The knowledge base is currently limited to 12 crops and Maharashtra-specific advisories; generalization to other regions requires corpus expansion.
- Mistral-7B inference requires a GPU for sub-3-second latency; CPU-only deployment is significantly slower (~18 s).
- The pest checker's accuracy degrades under low-resolution or poorly lit field photographs compared to controlled lab images.
- Voice input support is absent, limiting accessibility for low- literacy farmers.

3. Future Work

- Multilingual support (Marathi, Hindi, Kannada) using multilingual SentenceTransformer variants and fine-tuned Indic LLMs.
- Integration of satellite-based soil and vegetation indices (NDVI, EVI) via the Sentinel-2 API for remote sensing-augmented recommendations.
- Mobile application with offline capability for use in low- connectivity rural environments.
- Federated learning across multiple district-level deployments to improve pest detection under diverse field conditions while preserving data privacy.
- Real-time market price integration to advise farmers on the commercial viability of crop selection decisions.

References

M. S. Swaminathan, "Bridging the knowledge gap in agriculture: The role of information technology," Current Science, vol. 90, no. 5, pp. 635–641, Mar. 2006.

P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 9459– 9474, 2020.

R. A. Plant and G. E. Stone, "Knowledge-based systems in agriculture," Advances in Agronomy, vol. 44, pp. 37–89, 1991.

A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, Apr. 2018.

G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," in Proc. 16th Conf. European Chapter of the Association for Computational Linguistics (EACL), Apr. 2021, pp. 874–880.

A. Q. Jiang et al., "Mistral 7B," arXiv preprint arXiv:2310.06825, Oct. 2023.

D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060, Nov. 2015.

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, Sep. 2016.

A. Crane-Droesch et al., "Heterogeneous adoption of a weather-aware crop advisory app: Implications for digital extension," American Journal of Agricultural Economics, vol. 101, no. 3, pp. 847–868, May 2019.

N. Reimers and I. Gurevych, "Sentence-BERT:

Sentence embeddings using Siamese BERT-networks," in Proc. 2019 Conf. Empirical Methods in Natural Language Processing (EMNLP), Nov. 2019, pp. 3982–3992.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in Proc. 40th Annual Meeting of the Association for Computational Linguistics (ACL), Jul. 2002, pp. 311–318.

ICAR, "Crop Production Guide," Indian Council of Agricultural Research, New Delhi, India, 2022. [Online]. Available: https://icar.org.in

OpenWeatherMap, "Current weather and forecast API documentation," 2024. [Online]. Available: https://openweathermap.org/api