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

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155127

Krishi Mitra: A Multilingual AI-Powered Conversational Agent for Indian Farmers Integrating Government Schemes, Real-Time Mandi Prices, Crop Disease Detection, and Agricultural Advisories.

Jayashri Gajre[1], Sujay Manasubdar[2], Ishan Chaudhari[3], Shubham Panigrahi[4], Suraj Patil[5]

Department of Electronics & Computer Science, Shah & Anchor Kutchhi Engineering College[1]

Students, Department of Electronics & Computer Science, Shah & Anchor Kutchhi Engineering College[2345]

Abstract: Agriculture remains the backbone of India's economy, employing approximately 55% of the total workforce. Yet the majority of Indian farmers – particularly those in rural areas – continue to face significant challenges in accessing timely, accurate, and language-appropriate information about government welfare schemes, crop insurance policies, real-time market prices, and crop disease management. Existing digital solutions are fragmented, language-exclusive (primarily English or Hindi), and fail to deliver the holistic advisory support that smallholder farmers require. This paper presents Krishi Mitra ("Farmer's Friend"), a multilingual, AI-powered conversational chatbot designed specifically to bridge this information gap. The system integrates a Retrieval-Augmented Generation (RAG) pipeline using ChromaDB as a vector store and the paraphrase-multilingual-MiniLM-L12-v2 sentence-transformer model for semantic embedding. Natural language responses are generated using the Llama-3.3-70b-versatile model served through the Groq API. The chatbot supports three languages – English, Hindi, and Marathi – with automatic language detection and cross-lingual translation performed via LLM prompting. Additionally, the system incorporates a real-time mandi (agricultural market) price retrieval module fetching live data from government portals, a Convolutional Neural Network (CNN)-based crop disease prediction sub-system using leaf image classification, and a Text-to-Speech (TTS) module powered by Google Text-to-Speech (gTTS) for audio accessibility. Deployed as a Flask REST API with a React.js frontend, the system was evaluated across multiple agricultural query categories, demonstrating high semantic relevance, language fidelity, and user accessibility.

Index Terms: Artificial Intelligence, Chatbot, Retrieval-Augmented Generation, Multilingual NLP, Indian Agriculture, Farmer Advisory, ChromaDB, Crop Disease Detection, Mandi Prices, Sentence Transformers, Groq, LLM.

I. INTRODUCTION

India is home to more than 100 million agricultural households, yet information asymmetry between government institutions and farming communities remains a persistent structural challenge [1]. The Indian government operates numerous welfare programmes including the Pradhan Mantri Fasal Bima Yojana (PMFBY), PM-KISAN, the Kisan Credit Card (KCC) scheme, and Soil Health Card distribution initiatives; however, awareness and enrolment rates among marginal and small farmers remain disproportionately low [2].

The proliferation of smartphones even in rural India – with active internet connections surpassing 700 million as of 2023 [3] – presents an unprecedented opportunity to deliver agricultural advisories via conversational AI. However, most existing platforms operate exclusively in English, while more than 60% of Indian farmers are literate primarily in regional languages such as Hindi, Marathi, Telugu, Bengali, and Kannada [4].

Krishi Mitra addresses these limitations by providing a unified, conversational, multilingual interface that consolidates curated knowledge on government agricultural schemes, live crop mandi prices sourced from official government data portals, insurance and subsidy advisories, CNN-based crop disease detection from leaf images, and voice-enabled TTS output for farmers with low literacy.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155127

The name Krishi Mitra derives from Sanskrit roots meaning "Farmer's Friend," reflecting the system's core objective of democratising agricultural knowledge. The remainder of this paper is structured as follows: Section II reviews related literature; Section III describes the system architecture; Section IV details the methodology; Section V presents implementation details; Section VI discusses experimental results; Section VII outlines limitations and future work; and Section VIII concludes the paper.

II. LITERATURE REVIEW

A. Conversational AI in Agriculture

Prior work on agricultural chatbots has predominantly focused on rule-based or keyword-matching systems. Jha et al. [5] developed a FAQ-based agri-advisory system for the Indian market but noted its inability to handle open-domain queries. More recent systems leverage transformer-based models: Ramesh et al. [6] proposed a BERT-based classifier for crop disease symptom queries, though it did not support multilingual input.

B. Retrieval-Augmented Generation (RAG)

Lewis et al. [7] introduced the RAG paradigm, combining dense retrieval with a seq2seq generator, demonstrating that grounding LLM responses in retrieved documents substantially reduces hallucination. Subsequent work by Gao et al. [8] showed that RAG systems outperform purely parametric LLMs on domain-specific knowledge tasks – a finding particularly relevant to the agricultural domain where factual accuracy is critical.

C. Multilingual NLP for Low-Resource Indian Languages

The paraphrase-multilingual-MiniLM-L12-v2 model [9] has demonstrated competitive cross-lingual performance on semantic similarity benchmarks across 50+ languages. Studies by Kakwani et al. [10] on IndicNLP show that multilingual pre-training substantially improves downstream performance for Indic scripts. The "translate-then-retrieve" strategy – translating queries to a pivot language (English) before retrieval – has been validated by Shi et al. [11] as an effective zero-shot multilingual retrieval approach.

D. Crop Disease Detection using CNN

Deep learning-based plant disease detection has matured significantly since Mohanty et al. [12] demonstrated over 99% accuracy on the PlantVillage dataset using AlexNet. More recent architectures including ResNet-50 and MobileNet have been applied to Indian crop varieties (rice, wheat, cotton, sugarcane) with notable accuracy gains in transfer learning regimes [13].

E. Agricultural Market Price Systems

Real-time commodity price dissemination has been explored through SMS-based platforms (e.g., Reuters Market Light, IKSL) and web portals. The Government of India's Agmarknet portal and the eNAM platform provide structured commodity price APIs that have been integrated in academic prototypes [14], though no published work unifies these with an AI conversational interface.

III. SYSTEM ARCHITECTURE

A. High-Level Architecture

Krishi Mitra follows a three-tier architecture consisting of a Presentation Layer (React.js frontend), an Application Layer (Flask REST API), and a Data and Intelligence Layer (ChromaDB vector store, Groq-hosted LLM, CNN disease model, and government data APIs).

User queries enter through a React.js web interface supporting text input, voice recognition, and image upload. The Flask backend orchestrates query routing, language detection, retrieval, and response generation. Structured knowledge is stored in ChromaDB as dense vector embeddings. Real-time mandi price data is fetched from government portals. Crop disease prediction is served by a locally hosted CNN model.

B. Component Description

1) Frontend (React.js):

The frontend provides a responsive chat interface supporting multilingual text input, microphone-based speech input (Web Speech API), image upload for disease detection, and audio playback of TTS responses.

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

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155127

2) Flask REST API:

The backend exposes three primary endpoints: /api/chat (POST) for language detection, retrieval, and LLM-based response generation; /api/tts (POST) for text-to-MP3 conversion via gTTS; and /api/disease (POST) for CNN-based leaf disease classification.

3) ChromaDB Vector Store:

ChromaDB serves as the persistent vector database. The knowledge base is encoded using paraphrase-multilingual-MiniLM-L12-v2 embeddings (384-dimensional vectors). Cosine similarity is used as the retrieval metric via the HNSW index.

4) Groq-Hosted LLM:

The llama-3.3-70b-versatile model, accessed via the Groq inference API, handles language detection, cross-lingual translation, and natural language response generation within a structured prompt framework.

5) CNN Disease Detection Module:

A CNN trained on a curated Indian crop leaf dataset performs multi-class classification of crop diseases. Inference is served locally, keeping sensitive agricultural image data on-premises.

IV. METHODOLOGY

A. Knowledge Base Construction

A structured CSV knowledge base (schemes_kb.csv) was curated from official government sources including the Ministry of Agriculture & Farmers Welfare portal, PM-KISAN documentation, PMFBY policy documents, and Soil Health Card programme guidelines. Each record contains the question in English (Question_EN) as the embedding anchor, along with answers in English, Hindi, and Marathi (Answer_EN, Answer_HI, Answer_MR), and metadata covering scheme name, category, eligibility, and benefit amount. The dataset covers 12 major government schemes, 8 crop insurance products, general mandi guidance, and crop-specific advisories.

B. Data Ingestion Pipeline

On system startup, the ingest_data() function checks whether the ChromaDB collection farmer_schemes already contains documents. If empty, it reads the CSV, extracts English questions as primary documents, and stores the multilingual metadata (Listing 1).

```
def ingest_data(csv_file):
  collection = chroma_client\
    .get_or_create_collection(
    name="farmer_schemes",
    embedding_function=emb_fn,
    metadata={"hnsw": "cosine"})
  if collection.count() > 0:
    return True
  df = pd.read_csv(csv_file)
  documents = df["Question_EN"]\
            .tolist()
  metadatas = df.drop(
    columns=["Question_EN"])\
    .to_dict("records")
  ids = [f"id_{i}" for i
        in range(len(df))]
  collection.add(
    documents=documents,
    metadatas=metadatas,
    ids=ids)
  return True
```

Listing 1. Knowledge Base Ingestion (excerpt)

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

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155127

C. Query Processing Pipeline

Stage 1 – Language Detection: The user's input is passed to the Llama-3.3-70b model with a constrained prompt requesting only the ISO 639-1 two-letter code (en, hi, or mr). Temperature is set to 0.1 for determinism.

Stage 2 – Cross-Lingual Translation: If Hindi or Marathi is detected, the query is translated to English via a second LLM call. This "translate-then-retrieve" strategy ensures semantically equivalent English documents are found regardless of the user's input language.

Stage 3 – Semantic Retrieval: The English-normalised query is embedded using the sentence-transformer model. ChromaDB performs cosine similarity search; the top-k (default k=1) most relevant document is selected along with its metadata.

Stage 4 – Response Generation: The original user query and retrieved answer are passed to Llama-3.3-70b instructing it to answer only using retrieved context, in the user's language. Temperature is 0.7 with a 300-token limit (Listing 2).

```
def get_bot_response(query, lang):
  search_q = translate_to_english(
    query, lang)
  collection = chroma_client\
    .get_collection(
    name="farmer_schemes",
    embedding_function=emb_fn)
  coll_data = collection.get(
    include=["documents",
          "metadatas"])
  kb_emb = emb_fn(
    coll_data["documents"])
  q_emb = emb_fn([search_q])
  hits = semantic_search(
    torch.tensor(q_emb),
    torch.tensor(kb_emb),
    top_k=1)
  return generate_answer(
    query, retrieved_ans, lang)
```

Listing 2. Core Response Pipeline (excerpt)

D. Real-Time Mandi Price Module

The mandi price sub-system fetches commodity prices from the Government of India's data aggregation APIs (Agmarknet/data.gov.in) using scheduled HTTP requests with a configurable refresh interval (default: 60 minutes). Prices are stored in a local SQLite cache to reduce API dependency and enable offline degraded-mode operation.

E. Crop Disease Detection Module

A CNN was trained to classify 15 disease categories across four major Indian crops: paddy, wheat, tomato, and cotton. The model uses MobileNetV2 as a feature extractor (transfer learning from ImageNet) with the classification head: Global Average Pooling → Dense(256, ReLU) → Dropout(0.3) → Dense(15, Softmax).

The training dataset comprised 12,000 images (80/20 train-validation split) from PlantVillage augmented with field photographs from Maharashtra and Punjab. Training ran for 30 epochs with Adam optimiser ($lr=10^{-4}$), categorical cross-entropy loss, and data augmentation (±20° rotation, horizontal flip, brightness jitter). Final validation accuracy: 93.7%.

F. Text-to-Speech Module

The TTS sub-system converts generated text to MP3 audio using gTTS (Google Text-to-Speech). The detected language code is passed directly to gTTS for phonetically correct Hindi and Marathi pronunciation. Audio is streamed as audio/mpeg without disk persistence to minimise latency.

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

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155127

V. IMPLEMENTATION

A. Technology Stack

TABLE I TECHNOLOGY STACK OF KRISHI MITRA

Component	Technology / Library
Frontend	React.js, Web Speech API, Axios
Backend API	Python 3.10, Flask 3.x, Flask-CORS
Vector Database	ChromaDB (HNSW, cosine)
Embedding Model	paraphrase-multilingual-MiniLM-L12-v2
LLM	Llama-3.3-70b-versatile (Groq API)
Disease Detection	MobileNetV2 (TensorFlow/Keras)
Text-to-Speech	gTTS (Google TTS)
Data Storage	ChromaDB (persistent), SQLite (mandi)
Deployment	Flask dev server, extensible to Docker

B. API Endpoint Specification

```
@app.route("/api/chat",
        methods=["POST"])
def chat():
  data = request.json
  msg = data.get("message","")
  lang = detect_language(
        msg)
  resp = get_bot_response(
        msg, lang)
  return jsonify({
    "response": resp,
    "language": lang,
    "success": True })
```

Listing 3. Chat API Endpoint

```
@app.route("/api/tts",
        methods=["POST"])
def text_to_speech():
  data = request.json
  text = data.get("text", "")
  lang = data.get("lang","en")
  mp3_fp = io.BytesIO()
  tts = gTTS(text=text,
          lang=lang)
  tts.write_to_fp(mp3_fp)
  mp3_fp.seek(0)
  return send_file(mp3_fp,
    mimetype="audio/mpeg")
```

Listing 4. Text-to-Speech API Endpoint

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

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155127

VI. RESULTS AND DISCUSSION

A. System Requirements Specification Summary

The system was developed against these functional requirements: automatic language detection (English/Hindi/Marathi); contextually accurate knowledge-base-grounded responses; real-time mandi price retrieval; crop disease classification with ≥90% accuracy; text-to-audio conversion; and sub-5-second response for 95% of queries under normal load.

B. Knowledge Base Evaluation

The RAG pipeline was evaluated on 200 test queries across three languages. Retrieval accuracy was measured as the fraction of queries for which the top-1 retrieved document was judged relevant by domain experts. Table II compares results against a BM25 lexical baseline.

TABLE II RETRIEVAL ACCURACY BY LANGUAGE AND CATEGORY

Query Language	Proposed (RAG)	BM25 Baseline
English	94.0%	88.5%
Hindi	89.5%	61.2%
Marathi	87.3%	54.8%
Overall	91.4%	74.0%

C. Response Quality Evaluation

Five agricultural domain experts rated 60 sampled responses (20 per language) on a 5-point Likert scale for correctness, naturalness, and usefulness. Results are summarised in Table III.

TABLE III EXPERT RESPONSE QUALITY EVALUATION (MEAN ± SD)

Language	Correctness	Naturalness	Usefulness
English	4.6 ± 0.4	4.5 ± 0.5	4.7 ± 0.3
Hindi	4.3 ± 0.6	4.4 ± 0.5	4.5 ± 0.4
Marathi	4.1 ± 0.7	4.2 ± 0.6	4.3 ± 0.5

D. Crop Disease Detection Results

TABLE IV CROP DISEASE DETECTION PERFORMANCE

Crop	Accuracy	Precision	Recall
Paddy	94.8%	0.94	0.93
Wheat	93.1%	0.92	0.94
Tomato	95.2%	0.95	0.95
Cotton	91.9%	0.91	0.90
Overall	93.7%	0.93	0.93

E. Response Latency

Average end-to-end latency was measured over 500 test queries. English queries averaged 2.1 ± 0.4s; Hindi/Marathi queries requiring translation averaged 3.8 ± 0.7s; disease detection averaged 1.2 ± 0.3s; and TTS generation for a 50-word response averaged 0.9 ± 0.2s. All categories met the sub-5-second SRS requirement for 95% of queries.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.155127

F. Qualitative Observations

The system correctly handled code-switched inputs by defaulting to English retrieval. Hallucination was effectively suppressed – responses were consistently grounded in retrieved context, and the system returned a polite "not found" message when no relevant document was retrieved. Marathi TTS output received slightly lower naturalness ratings, suggesting room for improvement in gTTS Marathi prosody. The SQLite mandi price cache significantly reduced repeated API calls.

VII. LIMITATIONS AND FUTURE WORK

A. Current Limitations

The system supports only English, Hindi, and Marathi; extending to Tamil, Telugu, Bengali, Kannada, and Punjabi would substantially broaden impact. Government scheme details change seasonally, necessitating an automated scraping and re-ingestion pipeline. LLM inference via the Groq API and mandi price retrieval require active internet connectivity – an edge-deployable quantised LLM would improve rural accessibility. The CNN model covers only four crops; expanding to 20+ Indian crops would increase practical utility.

B. Future Directions

Future work will integrate Whisper-based ASR for voice-first hands-free field usage and deploy the system as a WhatsApp Business API chatbot. Weather forecast API (IMD, OpenMeteo) integration is planned for personalised crop advisories, alongside fine-tuning of a smaller domain-specific LLM (e.g., IndicBERT, MuRIL) for on-device inference. Federated learning across agricultural extension officer networks will be explored for privacy-preserving model improvement, and eNAM platform integration is targeted for direct price negotiation support.

VIII. CONCLUSION

This paper presented Krishi Mitra, a multilingual AI-powered agricultural advisory chatbot tailored for Indian farmers. The system integrates a RAG pipeline over a curated government scheme knowledge base, real-time mandi price retrieval, CNN-based crop disease detection, and multilingual TTS – all within a unified conversational interface via a React.js frontend and Flask REST API.

Key contributions include: a validated translate-then-retrieve pipeline achieving 91.4% overall top-1 retrieval accuracy, outperforming BM25 by 17.4 percentage points; a MobileNetV2-based crop disease classifier achieving 93.7% accuracy across four Indian crops; and an end-to-end architecture meeting sub-5-second latency requirements for 95% of queries. Development spanned two semesters: Semester 7 established the multilingual RAG infrastructure; Semester 8 added real-time market data and computer vision capabilities. Expert evaluation confirms that Krishi Mitra meaningfully advances AI-assisted agricultural advisory for the Indian context. Future work will expand language coverage, enable voice-first interaction, and deploy on WhatsApp to maximise farmer reach across rural India.

ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance of faculty mentors at Shah & Anchor Kutchhi Engineering College (SAKEC), Mumbai, and the Department of Electronics & Computer Engineering for providing laboratory infrastructure and research support. The authors also thank the agricultural extension officers of Maharashtra who volunteered their time for expert evaluation.

REFERENCES

[1] NITI Aayog, "Doubling Farmers' Income: Rationale, Strategy, Prospects and Action Plan," National Institution for Transforming India, New Delhi, Working Paper, 2018.
[2] Ministry of Agriculture & Farmers Welfare, Government of India, "PM-KISAN: Progress Report 2023," New Delhi, 2023. [Online]. Available: https://pmkisan.gov.in
[3] Telecom Regulatory Authority of India (TRAI), "Telecom Subscription Data," New Delhi, 2023. [Online]. Available: https://www.trai.gov.in
[4] Office of the Registrar General & Census Commissioner, India, "Census of India 2011: Literacy and Level of Education," New Delhi, 2011.
[5] S. Jha, A. Singh, and R. Sharma, "Development of an Agri-Advisory Chatbot for Indian Farmers using Natural Language Processing," in Proc. IEEE ICCCA, Greater Noida, India, 2019, pp. 1–6.

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.155127

[6] P. Ramesh, K. Vijayakumar, and S. Krishnamurthy, "BERT-based Crop Disease Query Classifier for Agricultural Advisory Systems," Computers and Electronics in Agriculture, vol. 198, pp. 107–116, 2022.

[7] 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.

[8] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv preprint arXiv:2312.10997, 2023.

[9] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, Hong Kong, 2019, pp. 3982–3992.

[10] D. Kakwani et al., "IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages," in Findings of EMNLP, 2020, pp. 4948–4961.

[11] W. Shi et al., "XRICL: Cross-lingual Retrieval-Augmented In-Context Learning," arXiv preprint arXiv:2210.13693, 2022.

[12] S. P. Mohanty, D. P. Hughes, and M. Salathe, "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.

[13] A. K. Rangarajan and R. Purushothaman, "Disease Classification of Mung Bean and Tomato Plants Using MobileNet," IEEE Access, vol. 8, pp. 186094–186108, 2020.

[14] Directorate of Marketing & Inspection, Ministry of Agriculture, Government of India, "Agmarknet: Agricultural Marketing Information Network," 2020. [Online]. Available: https://agmarknet.gov.in

[15] Meta AI, "Llama 3.3: Large Language Model Technical Report," 2024. [Online]. Available: https://ai.meta.com/llama/

[16] Groq Inc., "Groq API Documentation," 2024. [Online]. Available: https://console.groq.com/docs

[17] Chroma, "ChromaDB: The Open-Source Embedding Database," 2023. [Online]. Available: https://www.trychroma.com