Thesis Report

A Multimodal AI-Based Agricultural Assistance System: Enhancing Farmer Decision-Making Through Visual Question Answering (VQA) in Bangladesh

by

Promit Dey Sarker Arjan
24141134
Mrittika Devi Mati
24341241
Argha Basak
22101398
Saib Sadman Bari
22101668

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
January, 2026.

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Promit Dey Sarker Arjan
24141134

Mrittika Devi Mati
24341241

Argha Basak
22101398

Saib Sadman Bari
22101668

Approval

The thesis titled "*A Multimodal AI-Based Agricultural Assistance System: Enhancing Farmer Decision-Making Through Visual Question Answering (VQA) in Bangladesh*" submitted by:

1. Promit Dey Sarker Arjan (24141134)
2. Mrittika Devi Mati (24341241)
3. Argha Basak (22101398)
4. Saib Sadman Bari (22101668)

of Spring 2026 has been accepted as satisfactory in partial fulfillment of the requirements for the degree of B.Sc. in Computer Science in Spring, 2026.

Examining Committee:

Supervisor:
Dr. Amitabha Chakrabarty

Dr. Amitabha Chakrabarty
Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
Partha Bhoumik

Partha Bhoumik
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
Dr. Md. Golam Rabiul Alam

Dr. Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
Dr. Sadia Hamid Kazi

Dr. Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Agriculture plays a vital role in Bangladesh, employing approximately 43% of the population, yet many farmers lack formal education and expert guidance, leading to uninformed decision-making. This research explores the development of a multimodal Visual Question Answering (VQA)-based agricultural assistance system that integrates computer vision, natural language processing (NLP), and speech processing to provide real-time, voice-based responses in Bangla, reducing the immediate need for agricultural experts. Instead of relying solely on CNN-based classification, the system employs Vision–Language Models (VLMs) with visual encoders for crop disease understanding, a Large Language Model (LLM) for agricultural query answering, and speech models for Bangla voice interaction. The system offers an intuitive, offline-compatible mobile solution tailored for farmers in low-resource environments. A conversational agricultural diagnostic dataset was developed containing 96,003 expert-style diagnostic entries with 9,138 rice leaf images across six classes. A structured dataset covering Bangladesh's diverse crops, pests, and diseases ensures AI model optimization and practical usability. A comparative evaluation assesses the multimodal AI system (Image + Text + Voice) against traditional single-input models (Image-only, Text-only) based on diagnostic accuracy, response relevance, and user satisfaction. In the final offline pipeline, MobileNetV3-Small achieved 95.35% disease classification accuracy with stable mobile CPU latency (12–18 ms), Gemma 3–1B produced the strongest overall response quality among tested lightweight LLMs (METEOR = 0.1095, cosine similarity = 0.539, and BERTScore F1 0.70, BertScore Precision: 0.72, BertScore Recall: 0.69), and Whisper-Base enabled faster-than-real-time Bangla speech recognition (RTF = 0.7859, WER = 0.353). This approach minimizes dependence on agricultural experts, saving time and costs while positively impacting the economy. A farmer field survey (n = 30) using five evaluation parameters indicates strong real-world usability: 92% rated the system helpful/very helpful, 88% found it easy to use, 84% reported successful offline usage, 86% expressed satisfaction with response clarity, and 82% reported improved confidence in decision-making. Ultimately, this research seeks to bridge the digital divide in agriculture, empowering farmers with an AI-driven solution for real-time problem-solving and enhanced food security.

Keywords: AI-driven agricultural assistance, Visual Question Answering (VQA), Crop disease detection, Natural Language Processing (NLP), Computer vision, Speech processing, Low-resource environments, Multimodal AI, Offline agricultural solution, Food security

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

CV Computer Vision

LLM Large Language Models

NLP Natural Language Processing

VQA Visual Question Answering

ASR Automatic Speech Recognition

BERT Bidirectional Encoder Representations from Transformers

CLIP Contrastive Language–Image Pretraining

CNN Convolutional Neural Network

GPS Global Positioning System

GPT Generative Pre-trained Transformer

LLaVA Large Language and Vision Assistant

Mozilla TTS Open-source text-to-speech system

Pruning Removing unnecessary model weights

Quantization Reducing model precision for efficiency

Tacotron 2 Neural network architecture for speech synthesis

TTS Text-to-Speech

XLM-R Cross-lingual RoBERTa

Chapter 1

Introduction

1.1 Background

Agriculture is the backbone of Bangladesh. It gives work to a big part of our people and plays a major role in food security. But even though it is so important, farmers face many problems. In rural or remote places, they often do not get the expert advice they need. Many of them struggle to spot diseases in crops or understand climate risks. So, they make decisions based on guesswork, which affects their harvest and income.

The problems do not stop there. Language barriers, low literacy, and weak digital infrastructure make things worse. Many apps are in English, or they need strong internet which most farmers do not have. So, even if tech is growing fast, the benefits are not reaching everyone.

But now, with AI improving so quickly especially in areas like multimodal learning and Large Language Model (LLM), there is a real chance to change things. Artificial Intelligence (AI) that combines image understanding, Computer Vision (CV), Natural Language Processing (NLP), and even speech tech can help farmers get the right advice in the right way.

One exciting technology here is Visual Question and Answering (VQA). It can look at an image and answer questions about it. Imagine a farmer clicking a photo of a diseased leaf and asking, in Bangla, "What is wrong with this?" and getting a useful answer instantly that is the power of VQA.

So, this research aims to build something just like that a multimodal AI system for Bangladeshi farmers. It will take input from images, voice, or text all in Bangla and give back smart, real-time answers. Most importantly, it will work both online and offline. That means even if a farmer has no internet, they will still get some help. Offline responses will be simple and rule-based. But when there is internet, they will get a full conversation with more detailed answers.

The goal is to create something real, something that can actually help our farmers not just stay as an idea on paper.

1.2 Rational of the Study or Motivation

The main reason behind this work is very simple: our farmers need help. They need easy, smart tools in their own language to make better farming choices. But sadly, most of today's AI farming tools are not made for them. These tools are trained on Western data, need strong internet, and often speak in English or use only text or image not all together.

This does not work for the people in our villages. Most farmers are not tech-savvy. Many can not read well or feel uncomfortable while typing. That is why they need something different a system that listens to their Bangla voice, understands pictures they take of their crops, and replies in Bangla, too. And if the internet is not there, the system should still give basic support. That is fair.

There is one more motivation, research-wise. Other disciplines are applying AI using various kinds of data (like voice + image), but this has not been the case with agriculture, particularly not in Bengali. And there are practically no data that can actually demonstrate what the Bangladeshi crops and plant diseases look like, and how they are described by people. That is a big gap. No AI can be useful without local data. This study, therefore, desires to bridge those gaps by accomplishing three big things:

- Firstly, developing appropriate multimodal data containing simultaneous data regarding the utilization of image, text, and voice within a particular module or independently.
- Secondly, creating an LLM-driven offline-first mobile system. It implies that farms receive assistance even in the absence of the internet, using the help of simple Q&A classifiers.
- Thirdly, trying to see how far this system will go to support an online conversation without failing or falling back, since real people do not pose one question and exit.

In the end, this work is not just about building an AI tool. It is about including people who are usually left out of the digital world. It is about helping them grow better food, face climate change, and live a bit easier. So yes the impact is both technological and social.

1.3 Problem Statement

Agriculture plays a vital role in Bangladesh and supports the livelihoods of many people. However, farmers especially in rural areas often face challenges in identifying and addressing crop diseases in a timely manner. When a farmer notices symptoms such as discoloration or damage on plant leaves, the typical process involves contacting the "local agricultural office". This process is frequently delayed due to logistical constraints, lack of immediate expert availability, or poor communication infrastructure. As a result, valuable time is lost, leading to crop damage, financial loss, and reduced yield.

Current agricultural advisory systems in Bangladesh are not efficient enough to handle real-time or near-real-time responses. Moreover, many farmers have limited literacy, and the systems are not accessible in the Bangla language, making it harder for them to interact effectively. Furthermore, most farmers do not have access to modern, high-end devices or stable internet connections. Many of them use basic, low-cost smartphones and have limited technical literacy. Internet connectivity is another major limitation in remote regions, further slowing down access to expert knowledge or digital services. Therefore, there is a need for a language-friendly, offline, low-cost-device-compatible system that can provide fast, accurate, and accessible agricultural support directly to farmers in diagnosing plant diseases and receiving appropriate treatment guidance promptly in their own environment. Additionally, existing systems often fail to imitate natural human interaction or hesitation, quickly responding with a fallback when uncertain. This can reduce trust and user engagement. Therefore, the proposed system will include a confidence-based delay mechanism to simulate human-like thinking behavior before fallback responses. Offline fallback may include simple label-based responses, while richer interaction will be possible online.

Chapter 2

Literature Review

2.1 Preliminaries

In developing a new solution of a multimodal AI-based system of agricultural assistance oriented to the needs of Bangladesh farmers, a person will be forced to examine the key technologies that will be utilized as the foundation of the system. Visual Question Answering (VQA), Multimodal AI, Computer Vision, Bangali Natural Language Processing (NLP), Speech Technologies, Offline AI, and Dataset Development are the key aspects outlined here.

2.1.1 Visual Question Answering (VQA)

VQA is like kind of the engine of a system. It makes the system intelligent, which can see an image and respond to a question about the image. Imagine the following scenario: a farmer observes something suspicious on a leaf. He then takes a photo. Then asks, "What disease is this?"or "What to do now?"That's it. The AI starts working. It runs the image through a CNN. Then it reads the question using language models like BERT or GPT. Finally, it gives back an answer. Hopefully a useful one. It is not always perfect. But it works surprisingly well when tuned right.

2.1.2 Multimodal AI Systems

Multimodal AI means the system can handle different types of input at once. Image, voice, text. All together. And not get confused. For our work, it handles:

- Crop photos (usually taken with low-end phones),
- Questions in Bangla spoken or typed,
- Context info: crop type, season, maybe even GPS location.

Models like CLIP, Flamingo, and LLaVA help stitch these modes together. Basically, they let the system "see "and "read"at the same time. It's quite magical when it works smoothly.

2.1.3 Computer Vision for Crop Disease Detection

Computer Vision gives eyes to the system. It detects and classifies diseases from images. Since rural users mostly rely on budget smartphones, we're going for lightweight CNNs like MobileNetV2, EfficientNet-lite.
A few of our fundamental approaches include: :

- Basic classification to determine types of diseases,
- Identification of the problem by marking its location,
- And data augmentation, which is as if it were artificially creating more data to train better models.

Datasets such as PlantVillage are beneficial; however, they do not necessarily correspond to local crops or how the disease manifests itself in this area. We are also creating our own data.

2.1.4 Natural Language Processing (NLP) and Bangla LLMs

Now, for the text part, NLP allows the system to understand the Bangla language, typed and spoken both(after transcribing). It is not only about translation but also about understanding context. We are working with models such as BanglaBERT, BanglaGPT, and a few multi-lingual ones such as mBERT and XLM-R. They will be fine-tuned regarding agricultural data. Since most of the large models do not naturally understand the issues surrounding farming in Bangla without being trained.

2.1.5 Bangla ASR/STT and TTS

We know that not everyone can type, especially illiterate farmers. And that's exactly where speech comes in. We need to be able to:

- Involve Automatic Speech Recognition (ASR) to convert a Bangla speech into text,
- Next, write the response of the system and read aloud, with TTS .

The first will be assisted by such ASR models as Whisper or wav2vec 2.0. In the case of TTS, Tacotron 2 and Mozilla TTS TTS have the ability to make Bangla sounds more natural. But the bad thing is that we require this to work offline. That's likely a challenge, but it's possible.

2.1.6 Offline AI and Edge Deployment

Because rural connectivity is weak, the system must be offline-capable.
We're doing this by:

- Compressing the AI models using pruning and quantization,
- Deploying them with mobile-friendly frameworks like TensorFlow Lite or Pytorch Mobile,

- Using simple UIs with tools like Flutter that run smoothly even on older Androids,
- And caching common queries for speed.

This guarantees the access in remote areas without internet connection.

2.1.7 Dataset Development for Bangladeshi Agriculture

Most publicly available agricultural datasets are either overly generic or focused on image-only classification tasks and therefore do not adequately reflect the diagnostic challenges faced by Bangladeshi farmers. In particular, such datasets fail to capture region-specific disease characteristics, local farming practices, and the way farmers naturally describe crop problems in the Bangla language.
To address these limitations, we developed a custom agricultural conversational dataset tailored for Bangladeshi rice cultivation by adopting the following approach:

- Collecting rice leaf disease images from publicly available and reputable datasets that include samples relevant to South Asian and Bangladeshi agricultural conditions,
- Constructing structured multi-turn question–answer interactions in Bangla, designed to resemble real diagnostic conversations between farmers and agricultural experts,
- Collecting disease-related textual information through systematic web scraping and validating all datapoints through review by an external agricultural expert to ensure scientific accuracy and practical relevance.

The resulting region-specific dataset captures realistic symptom descriptions, diagnostic reasoning, and treatment recommendations commonly encountered in Bangladeshi agriculture. This dataset not only supports the objectives of the present study but also provides a valuable resource for future AI-based agricultural applications targeting underrepresented and low-resource farming communities.

2.2 Review of Existing Research

2.2.1 Crop Disease Detection

This category focuses on AI models designed to identify plant diseases from images, often emphasizing accuracy, efficiency, and deployment on mobile devices. A 67-layer hybrid model combining convolutional layers with fuzzy ReLU and Chaotic Particle Swarm Optimization was developed for cucumber leaf disease detection, achieving 97.28% accuracy and a small model size of 3.4MB suitable for mobile phones [34]. To address dataset imbalance, another study used DCGANs to augment a dataset from 34,000 to over 66,000 images, training a hybrid ResNet and DenseNet model to 97.8% accuracy and creating a WeChat mini-program for mobile use [45]. A comparative study of a custom CNN, AlexNet, and MobileNet on a dataset of 24,156 images found that MobileNet performed best, with 97.33% classification accuracy and 87.82% severity detection accuracy [17]. Similarly, the VGG-ICNN, a hybrid of

VGG16 and Inception-v7, reached 99.16% accuracy across 93 disease classes with only 6 million parameters [13]. For rice diseases, one paper used GANs to enrich a dataset, with a MobileNetV2 model achieving 98.54% accuracy and being deployed as a web-based tool [41]. Another rice disease study evaluated four CNNs, with DenseNet121 being the most accurate at 97.5% and MobileNetV3 being the most efficient [4]. For cotton plants, an ensemble method using Continuous Wavelet Transform (CWT) with VGG-19 reached 97.5% accuracy [24]. For nightshade crops, a lightweight CNN with advanced image segmentation achieved 95–100% accuracy [18]. Lastly, a model for mango fruit diseases used ResNet50 to estimate severity, achieving 97.82% test accuracy in a mobile app [35]. Common limitations across these studies include a lack of testing in uncontrolled real-world environments, an inability to estimate disease severity, and a need for better model interpretability.

2.2.2 Visual Question Answering (VQA) in Agriculture

VQA models in agriculture aim to provide interactive diagnostics by answering user questions based on images. One novel approach integrates deep learning and VQA through federated learning for wheat rust detection, developing custom datasets (WheatRustDL2024 and WheatRustVQA) and achieving 97.69% classification accuracy with a ResNet model and a BLEU score of 0.6235 with a BLIP-based VQA model [22]. Another study proposed the ILCD model, which uses coattention mechanisms and multimodal fusion to provide detailed textual descriptions of disease attributes, reaching 86.06% accuracy on its custom CDwPK-VQA dataset [29]. For fruit tree diseases, a VQA system using Tucker decomposition and a modular co-attention mechanism achieved 86.36% accuracy, offering a more practical and interactive tool for farmers [10]. A foundational paper introduced the "VQA-Machine," which leverages existing vision tools and a co-attention mechanism to answer questions and provide human-readable explanations [1]. To address the lack of data, the Crop Disease Domain Multimodal (CDDM) dataset was created, featuring 137,000 images and 1 million question-answer pairs, and was used to fine-tune a Large Vision-Language Model (LVLM) with a focus on enhancing the visual encoder to distinguish similar diseases [20]. Key limitations include models being confined to specific diseases or domains, struggling with unseen disease types, and having difficulty with abstract questions.

2.2.3 Chatbots and QA Systems

Chatbots and question-answering systems are being developed to bridge the knowledge gap for farmers by providing timely and accessible advice. "AgroBot" is a chatbot that handles both voice and text inputs, trained on over 8 million real-world queries to achieve 99.68% accuracy on topics from crops to weather [12]. Another system, AgriResponse, built a knowledge base from 5.9 million call logs and uses approximate matching to answer text queries with 85.43% accuracy [6]. For high-performance QA, Chains-BERT integrates textual relational chains into a BERT architecture, achieving state-of-the-art accuracy in agricultural scenarios [8]. More recent systems leverage large language models (LLMs); Farmer.Chat uses Retrieval-Augmented Generation (RAG) to provide localized advice on platforms like WhatsApp and has served over 15,000 users with a success rate greater than

75% [26]. The ChatAgri framework taps into ChatGPT's zero-shot and few-shot abilities to classify agricultural texts without domain-specific training, proving as effective as fine-tuned models [14]. Similarly, LLaVA-PlantDiag, a fine-tuned version of LLaVA, functions as a chat-style assistant for plant disease diagnosis, achieving 96% accuracy [25]. Common challenges include a limited domain of knowledge, a lack of robust multilingual support, and the need for real-world field testing.

2.2.4 Precision Agriculture and ICT

This category covers the broad application of Information and Communication Technology (ICT) and AI to optimize farming practices. A conceptual study based on a qualitative analysis of 29 documents found that accessible ICTs like mobile phones and community radios are highly effective for disseminating weather forecasts and crop advice to farmers, though their impact is limited by poor infrastructure, digital illiteracy, and cost [40]. In a more direct application, a method using high-resolution UAV imagery combined with vegetation indices and texture analysis was developed to diagnose salt stress in alfalfa crops, achieving 93.6% prediction accuracy [39]. A comprehensive review of AI in precision farming highlighted its use in crop and soil monitoring, pest control, and irrigation, with case studies showing AI models can detect crop problems with up to 97% accuracy [37]. The main barriers to wider adoption identified across these papers include the high cost of technology, the need for farmer training, and data integration challenges.

2.2.5 AI Frameworks and Deployment

This category focuses on the end-to-end process of building, deploying, and operationalizing AI in agriculture. The AgriUXE platform was proposed as a novel framework combining Explainable AI (XAI) with multimodal data sources to make agricultural insights transparent and trustworthy for farmers, with a proof-of-concept mobile app called AgriDash developed for viticulture [23]. To ensure accessibility in remote areas, a smart crop disease diagnosis system called LeafNet was developed with offline capabilities and multilingual support, designed to run on mobile devices and sync with the cloud when reconnected [31]. A real-time, web-based tool for rice disease diagnosis was also presented to make diagnostics more accessible [41]. Another paper detailed a full-stack workflow for deploying AI applications on resource-constrained edge devices, involving model training, post-training pruning, and optimization for the NVIDIA Jetson platform, with pruned models retaining over 95% accuracy [27]. Limitations in this area include the lack of formal accuracy evaluations and large-scale user studies for new platforms, challenges in maintaining accuracy after model pruning, and a reliance on specific hardware ecosystems.

2.2.6 Human-Centered AI and Policy

This area examines the integration of AI in agriculture from a human, cultural, and policy perspective. A conceptual paper on "Agriculture 5.0" advocates for a Human-Centered AI (HCAI) paradigm where technology complements human intuition and ethical judgment rather than replacing it [16]. Another study calls for culturally rooted digital interventions in African agriculture, proposing the "Plug-In Principle"

to ensure new ICT tools are designed to support and integrate with indigenous knowledge systems [42]. A separate paper explores the potential and pitfalls of using generative AI tools like DALL·E 2 for science communication, finding that while they have potential, they often lack scientific accuracy and rely on cliches [19]. The limitations of these approaches are that they are often theoretical and lack empirical data or concrete implementation models. Furthermore, there is no globally accepted framework for HCAI specifically for the agricultural sector.

2.2.7 Multimodal and Language-Specific Models

This category includes AI models that process multiple data types or are tailored for specific, often under-represented, languages. A practical transformer-based multimodal system was proposed that combines images, text, and sensor data for disease prediction and QA, achieving up to 96% accuracy and featuring a multilingual corpus and knowledge graph [21]. To address the gap in resources for the Bangla language, the ChitroJera VQA dataset was created with over 15,000 culturally relevant samples, and a novel dual-encoder model using BanglaBERT-BEiT was developed [15]. Similarly, the BVQA dataset was created for Bengali with over 17,800 question-answer pairs, and was used to train a new MCRAN model [33]. A bilingual OCR system was also detailed to recognize handwritten Bangla text and translate it for agricultural queries, designed for offline mobile use [30]. A domain-specific multimodal assistant was created by building a special dataset called Agri400K and using a two-stage fine-tuning process, significantly outperforming general models [28]. A different approach trained general models to generate domain-specific training material for a new model, AgroGPT, enabling it to handle complex conversations [32]. Finally, a conceptual paper discusses how Artificial General Intelligence (AGI) could integrate various data types to revolutionize agricultural decision-making [11]. A key challenge across these models is the high computational cost and the need for large, annotated datasets.

2.2.8 Image Captioning and Weed Detection

This category highlights specific AI applications focused on generating descriptions of images and identifying weeds. For weed detection, one study provided a practical solution by testing different YOLO models on 6,000 images taken in real-world conditions, achieving up to 91% accuracy with a processing time as low as 9.43 ms, making them suitable for real-time field systems [7]. This work underscores that using realistic data and lightweight models is a key strategy for progress in precision agriculture. For image captioning, the BLIP-DP model was designed specifically for plant diseases. Its innovative approach involves first using a VQA system to identify the disease, then using that information to build a dynamic text prompt to guide the language model, resulting in more accurate and relevant captions with a BLEU-4 score of 83.4 [38].

2.3 Summary of Key Findings

Table 2.1: Summary of Key Findings from Reviewed Literature

Paper Reference	**Key Contribution**	**Model or Technology Used**	**Accuracy/ Performance**	**Application/ Note**
[34]	Lightweight and accurate disease detection	CNN + Fuzzy Logic + Chaotic PSO	97%, 3.4MB model size	Mobile-friendly model for field use
[12]	Bangla voice/text AI chatbot trained on real farmer queries	NLP Chatbot	99.68%	Helps illiterate/semi-literate farmers
[45]	Solves class imbalance by generating synthetic images	ResNet + DenseNet + DCGAN	97.8%	Augmented dataset for fairer training
[17]	Lightweight plant disease model for phones	MobileNet	97.33%, 4.24M params	Budget Android compatible
[41]	Data augmentation with synthetic images	MobileNetV2 + CycleGAN	98.54%	Boosts real-synthetic training
[18]	Field-use app with simple interface	Lightweight CNN	99%	Offline use for farmers
[8]	Semi-supervised and contrastive learning for QA	Chains-BERT	Not specified	High QA performance
[6]	Large-scale question-answering system for farmers	Approximate Matching	85.43%	Based on 5.9M questions from KCC India
[22]	VQA model for plant disease diagnosis	Federated VQA Model	97.69%, BLEU 0.6235	Advanced VQA in agriculture

Paper Reference	Key Contribution	Model or Technology Used	Accuracy/ Performance	Application/ Note
[29]	Enhanced VQA with multimodal fusion	Co-attention VQA	86.06%	Improved plant disease QA
[15], [33]	Language-specific Bengali VQA models	BanglaBERT-BEiT	Outperformed general models	Regionally adapted for Bangla farmers
[28]	Vision-language model fine-tuned on agriculture	Agri-LLaVA	Better than general models	Based on 400K agri samples
[32]	Vision-language model trained in 3 expert stages	AgroGPT	Effective in long conversations	Domain-specialized ChatGPT-style system
[25]	GPT-trained QA model for plant pathology	LLaVA-PlantDiag	96%	Uses GPT-generated questions
[23]	Multimodal insights with explainable AI	IoT + XAI + Satellites	Not specified	Decision support on mobile
[20]	Vision transformers with multitask learning	PDLC-ViT	99.97% accuracy, 0.9973 IoU	High-precision localization of disease
[16]	Ethical, explainable AI in Agriculture 5.0	Human-in-the-loop design	Not applicable	Advocates for trust and collaboration

2.4 Research Gaps in Existing Studies

- Existing research on AI in agriculture is scattered and fragmented rather than unified.
- Most studies focus on single tasks (e.g., only crop disease detection or only Q&A), while very few integrate multiple functions into one smooth system.
- Many solutions are cloud-dependent and not properly tested in real rural/village field conditions, so performance drops when internet or power is weak.
- There is a major shortage of Bangla-language agricultural datasets, making many systems feel foreign and less useful for Bangladeshi farmers.

- These systems typically lack robust fallback strategies to handle uncertainty (e.g., admitting low confidence, asking clarifying questions, or escalating).
- Powerful multimodal models that could combine vision + language are usually too large and resource-heavy, making them impractical for mobile devices and offline/low-resource areas.

Chapter 3

Requirements, Impacts and Constraints

This chapter provides the main requirements, expected impact, and practical constraints on which the design and implementation of the proposed Bangla-based agricultural decision-support system will be guided. It specifies first the functional and non-functional requirements, which means that the solution will be based on the actual needs of farming and able to work in conditions of low resources and offline. It also includes the dataset construction and organization on which image-based disease recognition and multi-turn Bangla advisory conversations are trained and evaluated, and how the samples are organized and labelled to allow training and evaluation. The chapter proceeds to provide the values of the societal, environmental, and ethical factors that can be applied in the deployment of AI in agricultural advisory. Finally, it concludes the project management plan, significant risks, the mitigation strategies, and an economic view on whether the system is viable to develop and implement.

3.1 Final Specifications and Requirements

3.1.1 Functional Requirements

The proposed system should provide support for decision-making in crop disease identification and advisory services through the use of the Bangla language. It shall accept leaf/crop images captured or uploaded by the user, apply standard preprocessing, and generate a disease prediction using the trained vision model, presenting the identified class with an interpretable indication of confidence. The system will also accommodate voice-based interaction in Bangla, where user-speech is converted to Bangla text through an STT module, and users are given the option to enter queries either by speech or typed text. Depending on the user query and the session situation, specifically what disease is predicted, the fine-tuned language model will produce coherent, multi-turn responses in Bengali and contain symptoms, preventive advice, and recommended activities. The system shall operate in both online and offline settings, enabling local inference through locally hosted model services when network connectivity is unavailable, and it shall support logging of outputs and evaluation results for analysis and reporting in this thesis.

3.1.2 Non-Functional Requirements

Alongside the functional requirements, the proposed Bangla-based agricultural decision-support system should meet some non-functional requirements to make it practically usable in rural Bangladesh. Given that the target users usually have access to cheap smartphones and have unreliable Internet connectivity, the system should be usable in offline and low-resource settings. The application must be responsively interactive in nature, so that the image diagnosis, speech transcription, and the generation of Bangali responses must be able to run in an acceptable latency to be used in the field in real-time. Resource efficiency is thus a necessity, and the deployed components must be lightweight in terms of memory footprint, CPU, and storage space.
The system should also be resistant to variability of real-world input. Image-based diagnosis must be able to withstand typical field factors like uneven lighting, background confusion, blurriness, and partial blockage. Equally, the voice interface must be usable in moderate levels of noise and pronunciation differences, and the system must gracefully accept typing errors, for example, by permitting a retry or returning to text input. The interface should be designed to accommodate low literacy users, where simplicity, straightforward responses in the Bangla language, concise and practical answers, and optional voice response should be favored as the interface.
Non-functional constraints that are also important are privacy and safety. Other users should not be exposed to user images and voice records, as well as the history of conversations, and the system should reduce unnecessary data storage. Whenever offline, the data is supposed to be localized to the device to minimize chances of accidental disclosure. Lastly, the software design must be modular and maintainable so that individual parts (vision model, STT, LLM, TTS) can be updated or replaced without redesigning the entire pipeline to support further improvement and expansion to new crops or diseases.

3.1.3 Dataset Description

The available Visual Question Answering (VQA) datasets in the agricultural domain, including PlantVillageVQA and IP-VQA, are mostly established on the basis of image-based question-answer pairs in crop disease and pest detection. Although these datasets have been useful in driving forward the research in agricultural VQA, they significantly rely on the presence of images and are concerned with single-turn, image-grounded queries. Consequently, they fail to sufficiently reflect the realistic diagnostic procedure that is undertaken in agricultural extension services, where farmers tend to report crop problems verbally, and diagnosis is met through several iterations of clarification. Conversely, numerous existing and popular agricultural datasets are labeled with images of diseases or the presence of pests, but do not have the question-answer format needed to perform tasks of VQA. Besides, the datasets are often not accompanied by more detailed textual notes on the symptoms development, environmental factors, disease transmission, and practice of treatment in the language of farmers. This leaves a definite gap in agri-cultural VQA resources to facilitate conversational, text-based, and low-resource diagnostic situations, especially in the local languages. To solve this issue, we present a large-scale agricultural conversational dataset that involves natural language descriptions of symptoms, multi-turn question-answer dialogues, and expert-like disease diagnosis and advisory services. The dataset is specifically designed to support the devel-

opment and evaluation of domain-specific VQA and conversational AI models for agricultural decision support, with an emphasis on offline usability and low-compute deployment.

3.1.4 Dataset Construction and Organization

Our custom annotated dataset is tailored for crop disease classification and conversational question–answering tasks in agriculture. The dataset contains a total of 96,003 diagnostic entries, of which 9,138 entries are associated with crop images, while the remaining entries are fully functional in text-only form. This design ensures that the dataset can be used in both multimodal and non-visual settings. Each dataset entry represents a complete diagnostic scenario and includes structured information such as the crop type, farmer-reported symptoms written in Bangla, a multi-turn question-answer flow, a confirmed disease label, and actionable treatment recommendations. Disease knowledge and advisory content are grounded in curated disease profiles developed using standard agronomic references and expert-reviewed information. The dataset is organised in a structured format, where crop images are stored in dedicated folders, and corresponding metadata, including symptom descriptions, questions, answers, disease labels, and interaction flow, are stored in structured data files. This separation allows flexible usage for training, evaluation, and deployment of conversational or multimodal models.

3.1.5 Disease Coverage

The dataset is concentrated on the diseases of rice crop including a representative set of fungus and virus diseases, and a healthy reference class. These categories of diseases are included: • Rice Blast (ধানের ব্লাস্ট) • Brown Spot (ব্রাউন স্পট) • Sheath Blight (ধানের খোলাপোড়া) • Leaf Scald / Leaf Scaled (পাতার আঁশযুক্ত রোগ) • Rice Tungro (ধানের টুঙ্গরো) • Healthy Paddy Leaf (সুস্থ ধানের পাতা) • Not Paddy (এলোপাতাড়ি ছবি) These classes cover the range of different degrees of severity and agronomic effects so that it is possible to assess the diagnostic reasoning and advisory accuracy.

3.1.6 Question–Answer Design

To every diagnostic entry, there is a structured multi-turn question-answer dialog, which is aimed at modeling realistic interactions between farmers and agricultural experts. As an alternative to using the fixed templates, the conversation flow is being developed by clarifying the questions, which concentrate on the observable characteristics like the shape of the lesion, color, the pattern of its spread, parts of the plant affected, and the course of the symptoms. The final response in each interaction includes:

- Disease identification
- Treatment recommendations (chemical, biological, or management-based)
- Additional preventive or contextual information

3.1.7 Sample Images from Custom Dataset

Field Background

Brown Spot

Leaf Scaled

Rice Blast

Rice Tungro

Sheath Blight

Figure 3.1: Sample images from the custom dataset (Field Background), rendered in a fixed square format.

White Background

Brown Spot

Leaf Scaled

Rice Blast

Rice Tungro

Sheath Blight

Figure 3.2: Sample images from the custom dataset (White Background), rendered in a fixed square format.

3.1.8 Example Conversational Flow

User: আমার ধানের পাতায় ছোট ছোট বাদামী রঙের দাগ দেখা যাচ্ছে। প্রথমে মনে হয়েছিল পোকা লেগেছে, কিন্তু দাগগুলো বাড়ছে।
Model: আপনার ধানের পাতায় কী ধরনের দাগ দেখা যাচ্ছে? দাগগুলো কি চোখের মতো গোল, নাকি অন্য কোনো আকারের? এবং কি দাগগুলো পুরো পাতায় ছড়িয়ে পড়ছে?
User: দাগগুলো প্রথমে ছোট ছিল, এখন অনেকটা চোখের মতো দেখাচ্ছে। চারপাশে হালকা বাদামী আর ভেতরে ধূসর। পাতাগুলো ধীরে ধীরে শুকিয়ে যাচ্ছে।
Model: আপনি কি গাছের কাণ্ড বা শীষেও কোনো দাগ বা পরিবর্তন লক্ষ্য করেছেন? যেমন, কাণ্ডের গিঁটে বা শীষের গোড়ায় কোনো দাগ?
User: হ্যাঁ, শীষের গোড়ায় একটু কালচে দেখাচ্ছে। মনে হচ্ছে শীষ বেরোনোর সময় ভেঙে যেতে পারে।
Model: রোগ নির্ণয়: Rice Blast (ধানের ব্লাস্ট)

আপনার ধানের জমিতে ব্লাস্ট রোগ দেখা দিয়েছে। দ্রুত Tricyclazole বা Propiconazole স্প্রে করুন। ইউরিয়ার ব্যবহার কমিয়ে দিন এবং জমিতে সবসময় পর্যাপ্ত পানি রাখুন।
অতিরিক্ত তথ্য: ব্লাস্ট রোগ ধানের জন্য খুবই ক্ষতিকর। এটি দ্রুত ছড়াতে পারে এবং ফলন কমিয়ে দিতে পারে। রোগ প্রতিরোধের জন্য রোগমুক্ত বীজ ব্যবহার করা উচিত।

3.1.9 Annotation Process and Quality Control

A systematic, designed annotation process was used to construct the dataset based on the reality of self-realistic interactions of agricultural diagnostics. The initial agronomic sources and extension sources used to obtain the original disease profiles were expert-curated. Natural language descriptions and multi-turn messages with questions and answers were created with the help of these profiles in order to simulate the dialogue between farmers and agricultural specialists. A match of every entry to a diagnosed disease and treatment possibilities was conducted to ensure consistency and diagnostic validity. The transcripts of the discussions were considered in accordance with the logical flow of the considerations of reporting the symptoms and answering the questions of clarification and the final diagnosis. This was done to ensure that the dataset is not only linguistically natural but also agronomically sound and would hence be used in training and testing conversational agricultural assistant systems.

3.1.10 Contribution and Use Cases

By integrating conversational depth, Bangla language support, and optional multimodal inputs, the proposed dataset bridges the gap between existing agricultural VQA datasets and real-world farming scenarios. It is most appropriate in training and testing agricultural assistants on mobile devices in rural settings where people have limited internet access and expert reach.

3.1.11 Dataset Preprocessing

In the dataset preprocessing stage, various measures were implemented to be able to guarantee the quality, consistency, and usability of the data to be used in supervised learning and conversational agricultural question-answering applications. First of all, we have checked that there are no duplicate images and that all the

image files mentioned in the dataset are there. We also performed a check on the metadata to identify that there are no entries with missing or conflicting information. All the crop images were then standardized to 244 x 244 pixels and then trained. This standardization process guarantees that there are consistent input dimensions for vision and multimodal models. and it also decreases the complexity of computation and consumption of memory during training. The process was followed by image normalization and data integrity verification, followed by a data split into a training and test set in an approximation of an 80-20 split, as is traditionally done when training supervised learning systems. Table 3.1 indicates the distribution of the images and question-answer pairs of the three subsets. The disease classes in each split were allocated in a well-planned manner so that all the disease categories were evenly distributed in both the training and test. We also made sure that in splitting the dataset, the entire multi-turn conversational structure of each data entry was maintained in the process of image-level preprocessing. All samples were left with their entire series of question-answer interactions in terms of farmer symptom descriptions, clarification questions, diagnostic conclusions, and treatment recommendations. This ensures that the context of conversation is not lost and can be used in the training and evaluation of multi-turn agricultural assistant models. Overall, the preprocessing pipeline results in a clean, balanced, and standardized dataset suitable for training text-only, multimodal, and conversational agricultural decision-support systems, particularly in offline and low-resource deployment environments.

Table 3.1: Train and Test set distributions of the dataset

Subset	**No. of Images**	**No. of Conversations**
Train	7,308	76,802
Test	1830	19200
Total	9,138	96,003

3.1.12 Data Collection

The process of data collection of the proposed agricultural conversational dataset was structured in such a manner that it would be diverse, scientifically accurate, and relevant to the real world as far as diagnosing rice disease in the Bangladesh agricultural settings. The dataset is a combination of visual and textual diagnostic datapoints (images), which have been originally gathered by a variety of trustworthy sources and then curated and validated.

3.1.13 Image Collection

To make the data transparent, reproducible, and ethically use, only publicly available data sets and widely used datasets were used to get rice leaf images. The primary image sources are:

- Dhan-Shomadhan: A Dataset of Rice Leaf Disease Classification for Bangladeshi Local Rice [3], hosted on Mendeley Data.
- Rice Leafs Disease Dataset (2023)[5], published on Kaggle by dedeikhsandwisaputra.
- Rice Leaf Disease Image Dataset (2020)[2], published on Kaggle by nirmalsankalana.

A total of 9,138 different images of rice leaves were chosen out of these sources. In the process of selection, the cases containing duplicate images, low-quality samples, and visually difficult cases were filtered out to ensure consistency and clarity of the data sets.

3.1.14 Textual Data Collection

The textual datapoints found in the dataset, including the descriptions of the symptoms, the nature of the disease, its causes, the methods of its spread, the environmental factors, and treatment advice, have been obtained due to systematic web scraping of various sources of agricultural knowledge. Such sources are Web-based agricultural extension resources, scientific literature, and publicly accessible portals of crop disease information. Information collected was later reorganized and rewritten in Bengali, with patterns of language that are farmer-focused, in order to capture the experience of the farmers in terms of describing crop problems in their natural language. This was necessary to make them usable in conversational agricultural assistant systems, specifically to serve rural users.

3.1.15 Dataset Integration

After image selection and text verification, the visual and textual components were integrated to form complete diagnostic entries. Each entry combines:

- A farmer-style symptom description in Bangla
- A structured multi-turn question–answer interaction
- A verified disease label
- Actionable treatment and prevention advice

This unification leads to a dataset with 96,003 diagnostic entries, 9,138 image-related samples, which would be appropriate for multimodal as well as text-only agricultural VQA and conversational tasks.

3.1.16 Comparison of Agricultural VQA Datasets

The comparison of the representative agricultural visual question answering (VQA) datasets and our proposed dataset is given in Table 3.2. The available datasets, including PlantVillageVQA and IP-VQA, have been significant in promoting the field of agricultural VQA research through the provision of image-based grounded question-answer pairs to analyze crop diseases and pests. Nevertheless, they are mostly image-based datasets and are mainly intended to solve one-turn or image-related VQA tasks. PlantVillageVQA is based on the popular PlantVillage image corpus and is a massive dataset of pairs of questions and answers concerning various crop species and disease circumstances. The dataset not only categorises questions into various vision-language model benchmarking, but it also classifies the questions into diverse cognitive levels, thus they can be used to benchmark vision-language models to identify plant diseases. On the same note, the IP-VQA dataset is dedicated to pest and crop health detection of insects in precision agriculture settings. It combines visual information with open-ended pairs of questions to help in multimodal analysis of agriculture. Nonetheless, as stated in the original publication of IP-VQA and its official sources, the number of images and question-answer pairs is not directly defined, and the dataset is rather characterized in terms of its implementation and VQA design based on images. Notwithstanding their efforts, both PlantVillageVQA and IP-VQA are dependent on the image availability and multimodal computation. In practical agricultural contexts, especially low-resource and rural settings, farmers tend to describe agricultural issues in a more natural language and not necessarily accompanied by images and diagnoses, and often undergo several rounds of clarification rather than a single question-answer dialogue. In contrast, our dataset is designed to model multi-turn, conversational agricultural diagnosis, emphasizing text-based symptom descriptions in Bangla while also supporting optional image inputs. The dataset contains 96,003 complete diagnostic entries and 9,138 associated images, making it suitable for both text-only and multimodal use cases. Each entry represents a realistic farmer-assistant interaction, including an initial symptom description, follow-up clarification questions, a verified disease label, and actionable advisory recommendations. The dataset can be used in open-ended and closed-ended queries, providing more contextual insights and performance when conversational agricultural assistant systems are required, especially in offline and low-compute contexts.

Table 3.2: Comparison of representative agricultural VQA datasets with our proposed dataset

Corpus	**Images**	**QA Dialogues**	**Domain Focus**	**Query Types**
PlantVillageVQA	55,448	193,609	Crop Diseases	Open-ended
IP-VQA	Image-based	Image-based QA (exact counts not specified)	Insect pests & crop health	Open-ended
Our Dataset	9,138	96,003 (conversations)	Rice Crop Diseases	Open-ended, Close-ended (multi-turn)

3.2 Societal Impact

The proposed offline-capable, Bangali-grounded VQA agricultural assistance system has a significant societal impact as it increases accessibility, equity, and timeliness of agricultural information to rural farming communities. The system will incorporate speech-to-text, a CNN-based disease detecting module, and a locally deployed LLM-based reasoning system to decrease the use of slow institutional support and physical visits to agricultural offices that may not be easily accessible to farmers because of the need to travel to the offices and the restricted access to experts. The voice-based interface of the system reduces the barriers of literacy and technology and allows even farmers with little formal education to implement practical and actionable information in a language that they can understand well. The modular offline-first workflow ensures continued usability under limited connectivity and low-end device constraints, supporting inclusive adoption while also strengthening privacy because farmer queries and images can remain within the local device ecosystem. The system assists small-scale farmers in making quicker judgments, promptly reacting to indicators of illness, and decreasing preventable losses in their crops, which leads to enhanced livelihoods and resiliency in small-scale farming contexts by providing instant diagnostic assistance and follow-up suggestions based on refined agricultural expertise.

3.3 Environmental Impact

In the context of environmental sustainability, the proposed system will help in sustainable farming practices, as it will be used to identify crop diseases early and correctly, thereby minimizing unnecessary application of pesticides and fertilisers. The ability to diagnose the problem in time will help the farmers to implement specific treatments instead of implementing generalized chemical interventions that may negatively affect the quality of the soil, contaminate water sources, and destroy the ecosystems around them. The lightweight model architecture and offline-first design reduce the energy usage, as the system does not need constant communication with the cloud, extensive computation, and is therefore environmentally friendly in terms of energy usage, as well as cost-effective. Also, the system facilitates accountable resource use and minimizes crop wastage due to delayed or wrong intervention by promoting localized decision-making on the basis of image evidence and contextual reasoning on the dataset. Such resource-aware AI-assisted farming devices can help in the long run to facilitate environmentally conscious agricultural methods and a balance between productivity and environmental conservation in resource-constrained rural settings.

3.4 Ethical Issues

The textual data was compiled based on various online resources and, therefore, all the points of the collected data were subjected to manual verification by an independent agricultural expert. The specialist checked the disease names, symptoms description, causal agents, treatment guidelines, and prevention to ensure scientific accuracy, agronomic soundness, and practical utility. This expert measure served

to eradicate the presence of unreliable, outdated, or misleading data added in the process of web scraping and to make sure that the dataset is in line with conventional agricultural practices. In addition, during field-level verification, farmers were consulted to assess whether the generated guidance is understandable and practically applicable in real farming contexts; informed consent was obtained from all participating farmers before conducting these verification activities. In addition, the photos that were taken with farmers and the voice recordings that were made during verification were acquired only with express permission. Such materials were to be applied only to research and evaluation and will not be published or made public. In cases where data were to be analyzed by documentation, the use of personal identifiable information was shunned, and the data were managed in such a manner that the privacy of study participants would be safeguarded and ethical research practices would be observed.

3.5 Summary

This chapter presented the proposed agricultural conversational dataset for rice leaf disease diagnosis. The data comprises 96, 003 diagnostic records, 9,138 rice leaf images, seven classes with five popular rice diseases, and a reference and not-paddy (random pictures) class. In every entry, there is a realistic diagnostic situation where a case of farmer-reported symptoms in Bangali is succeeded by a structured multi-turn interaction of question and answer and a series of advisory responses in an expert fashion. The dataset combines visual and textual data: pictures were taken from publicly available repositories with rice disease, and textual information about the disease was systematically stacked using web scraping, and afterwards, an external agricultural scientist was used to verify that the data is scientifically accurate and practically valuable. To facilitate supervised learning and equitable assessment, the dataset was preprocessed and divided into smaller segments using a stratified approach strategy to attain balanced representation of the disease classes, and all the images were resized to 244 x 244 pixels before being trained. This data serves as a basis for the model fine-tuning and evaluation in the following chapters, enabling the application of this model in multimodal and text-only decision support tasks in low-resource and offline scenarios.

3.6 Project Management Plan

The implementation of the research was carried out based on a milestone plan to make sure there was orderly progression in the areas of dataset development, training of models, system integration, and evaluation. The work began with the analysis of the requirements and scope description, then came the design of the data schema and annotation rules. The data were collected simultaneously with (i) rice-leaf pictures and (ii) textual data related to the disease, after which they were preprocessed and formatted. The dataset content was then validated by an external agricultural expert, and farmer consultations were conducted with informed consent to confirm practical relevance. Next, baseline vision and language models were selected and fine-tuned using the prepared dataset. The quantitative measures and qualitative output evaluation were used to test the performance of models. Lastly, the trained

modules were added to the proposed pipeline (image-based diagnosis, Bangla STT, and conversational response generation), and the research was completed with the field-oriented testing, documentation, and limitation and future work-related reporting. Periodic progress tracking and version control had been sustained through the research.

3.7 Risk Management

The main risks of this research are the wrong diagnosis of the disease or inappropriate suggestions, which can lead to a bad decision and even the loss of crops. To mitigate this, the system incorporates uncertainty-aware responses, prompts users for clearer inputs when necessary, and frames outputs as decision support rather than expert replacement. The risks associated with datasets, including untrustworthy or out-of-date web-scraped data, were minimized by using expert authentication and farmer consent on a given level. Generalization risk due to dataset imbalance or variable image conditions was addressed through careful curation, standardized preprocessing, and transparent reporting of limitations. Operational risks include reduced usability in limited computing environments in the offline mode, and transcription errors in Bangla STT are possible because of noise or dialect variation; these were alleviated by experimenting with varying model sizes, using a modular deployment strategy, and allowing review and re-try of transcription. The issue of privacy of user images, voice, and queries was addressed by reducing the amount of data stored and only logging what is required to be evaluated.

3.8 Economic Analysis

The proposed system offers a cost-effective decision-support approach by providing rapid Bangla guidance without requiring frequent in-person consultation. Development costs are primarily associated with dataset preparation (collection, preprocessing, expert review), model training resources, and deployment hardware for offline inference. In offline mode, recurring operational costs can be maintained at a low level during deployment since inference is carried out locally. It is expected that the benefits will be less time and traveling expenses to the farmers, sooner management of diseases, and better knowledge on preventive measures, which will indirectly reduce the loss of yield. Participants during farmer consultations stated that it can take several weeks to receive assistance from external sources (e.g., agricultural offices or local experts), and by the time they get the advice, the disease symptoms may have spread so much that intervention will be ineffective and the yield will be limited. The suggested system could minimize the delay of the response and allow the earlier action to be taken by offering real-time, offline-ready guidance and assisting farmers in avoiding avoidable losses, and improve economic resilience. All in all, the system is economically viable since the high cost is incurred in the process of development, yet the marginal cost per extra user is relatively cheap.

Chapter 4

Proposed Methodology

This chapter outlines the approach taken in the design process, implementation, and evaluation of the proposed AgroAssist system. It describes the entire process of end-to-end development, such as requirement-based planning, selection of the module to be offline deployed, and developing the multimodal pipeline that includes image-based disease recognition, Bangali speech-to-text, and conversational reply generation. The chapter also provides the initial system design, the training policy, and the hyperparameter values applied in fine-tuning the chosen LLM/VLM models, and an experimental environment such that model performance is evaluated in resource-constrained situations.

4.1 Workplan

In this chapter, the workflow that has been used to design and implement the AgroAssist system is presented in general. This research workplan was broken down into a number of organized stages, in which each stage was used in the fulfillment of the ultimate goal: a practical, offline, and Bangali-speaking agricultural assistant that promotes both the image-based disease support and voice-based interface. The implementation processes were conducted in a logical sequence where the system design and module selection were started, and finally, full pipeline testing and verification were done.

4.1.1 System Design and Requirement Planning

The initial stage was aimed at identifying the fundamental needs of the app according to the practical limitations in the world of agriculture. The system was to be an offline-first system due to the unreliable connectivity in most rural regions. The design was focused on multimodal interaction, meaning the farmers have an option to either give an image of a crop or talk in Bangla, explaining their problem. Other usability requirements of this phase included: short responses, simple language, voice output support to make the product easy to understand.

4.1.2 Module Selection and Technology Integration

Our definition of requirements then led to the selection of implementation modules that will be efficient when deployed in mobile devices. Flutter was selected as a

cross-platform developer and fast UI designer. To achieve on-device image-based understanding, a mobile-friendly vision classifier was installed using the PyTorch Lite interface. In the case of speech transcription, the pipeline was configured based on the local STT service that operates as a lightweight C++ server. In the case of conversational response generation, a local LLM service was added, also with a C++ server interface. A TTS engine was used to support Bangla voice output to make the assistant speak.

4.1.3 On-device Image Workflow Development

The second step involved the execution of the image workflow. Once the user feeds in a crop image, the system does preprocessing by decoding, resizing, and converting it into a mobile-compatible inference format in the form of tensors. The program executes on-device inference to determine the most probable disease category along with a level of confidence. The detected context is then stored in the active chat session, which enables the assistant to switch to a disease-aware mode where responses are confined to the detected problem and are consistent with follow-up questions.

4.1.4 Voice Interaction Pipeline Development

Parallel to it, the speech workflow was created to be used hands-free. The application captures the speech of farmers in WAV format and forwards it to the local STT (C++ server) to be transcribed. The transcription is made as a user message into the chat history as Bangali text. This allows a free-flowing conversation with the farmer being able to talk more and the assistant being able to reply, both textually and verbally. There is also the fail-safe processing of empty recordings, errors of transcription, and malfunctions of the connection between the app and the local servers.

4.1.5 Session Management and Persistent Memory

The system was developed to have persistent sessions to enable real-life use over several days. Every chat session contains the history of the messages, time, image association (if present), detected disease scenario, confidence rating, and follow-up counters. The feature of a long-term memory note was also introduced to remember the constant user context, such as. location of farming or preferences. All the session data and memory notes were locally stored in order to ensure that the system can be used in full offline mode and should be able to recover past conversations.

4.1.6 Prompt Control and Response Consistency Enforcement

To make the assistant act in a controlled and farmer-oriented way, a specific stage devoted to the development of response policy. The system dynamically adjusts its prompting rules based on context: image provided with detected disease, image provided without detection, or voice-only conversation without diagnosis. When the assistant is used in the voice-only mode, it poses one question at a time to detect the problem progressively. After the identification of a possible disease, the system

modifies the context of the session and proceeds in a disease-directed mode. Post-processing response was also used to remove bullet format, impose terse answers, and make each response complete with one follow-up question.

4.1.7 Testing, Verification, and Final Pipeline Test

Lastly, the entire system was tested with structured checks of every component of the system and the entire end-to-end flow. This phase confirmed the stability of on-device inference, the reliability of STT transcription, the reliability of the generation of responses by the LLM, TTS voice output, session persistence, and health monitoring of the server. To aid in diagnosing the problems caused by runtime, the application also contains the real-time server connectivity indicators of STT and LLM. This follows a systematic workflow that made the system reproducible, scalable, and capable of offline Bangla agricultural support in resource-challenged environments.

4.2 Preliminary Design or Design Specification

4.2.1 Proposed Model

The primary goal of the study was to come up with a light but efficient VQA framework that could be used to diagnose leaf diseases of crops through a visual input and natural language query. Even though recent multimodal systems have shown good performance during vision-language reasoning problems, they usually consume large parameter sizes and require high computational resources. These demands restrict their applicability to real-world agricultural contexts, especially in rural communities where internet connectivity and hardware developments are low. In order to address these limitations, we present a small-scale and compact architecture that is designed to be deployed offline and run on low-end devices. The system we propose is a hybrid modular system with visual perception and language-based reasoning being addressed by independent but coordinated components. This division decreases the computational costs and does not affect diagnostic quality and interactive behavior.

4.2.2 Image Encoding and Visual Diagnosis Module

To analyze the images visually, a light CNN image classification model was used to find the crop diseases by the use of leaf images. The input image is resized and normalized, then it is fed into the network, giving a one-dimensional logits vector. A maximum logit value is used to determine the predicted disease class, with a normalized confidence score being achieved using a softmax function. Figure 4.1 demonstrates that the model uses a typical CNN pipeline whereby the convolution and pooling layers are used to extract discriminatory features, which are flattened and sent to the fully connected classification layer to produce the final prediction. This module is designed to run on a device and runs completely offline. CNN, although having a small architecture, is an effective model in capturing discriminative patterns of visual data of different disease types, making it possible to recognize diseases with accuracy and speed without requiring GPU acceleration.

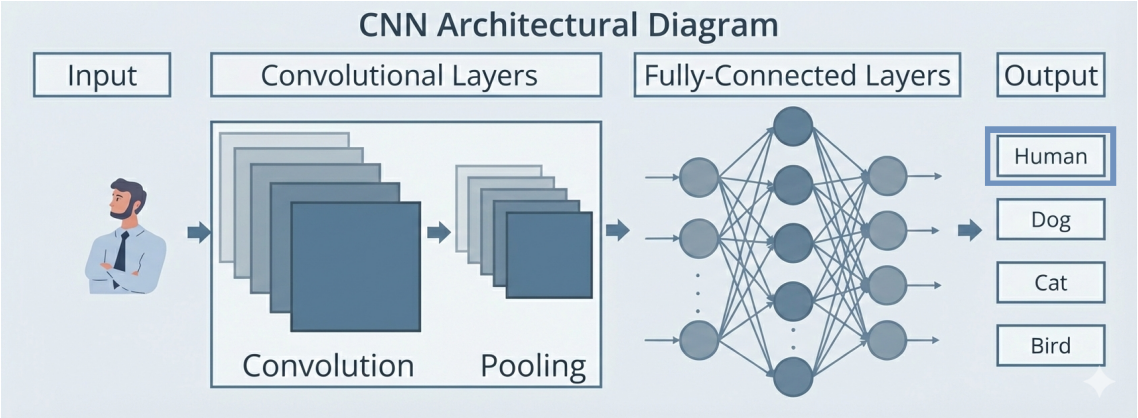

Figure 4.1: CNN Architecture

4.2.3 Speech and Text Processing Module

The system uses a speech-driven interface to facilitate natural and natural interaction with the farmers. The queries are received in the audio format and processed by a local speech processing server written in C++, which translates the verbal input into text. This will enable the system to be highly reliable in a low connectivity environment and, at the same time, retain its usability to farmers who might be less at ease with text-based input. The speech-to-text pipeline, as shown in Figure 4.2, converts the input waveform into acoustic features and decodes them in an encoder-decoder architecture to text tokens. In the case of verbal feedback, when it comes to the mobile application, the platform uses the default Android Text-to-Speech (TTS) engine that has been made available using Flutter, which allows the system to provide the voice response in Bangla and in full offline mode without any voice services in the cloud. Figure 4.3 demonstrates that the text-to-speech module transforms the generated text into speech output in three steps of normalization, acoustic modeling, and waveform synthesis. The text that is transcribed is also passed on to the language understanding component to give additional reasoning and a generated response.

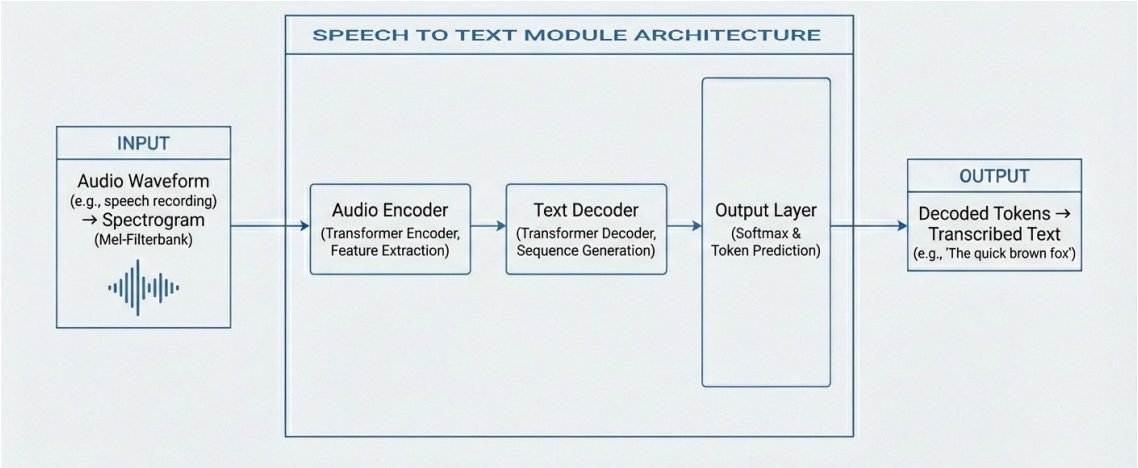

Figure 4.2: STT Architecture

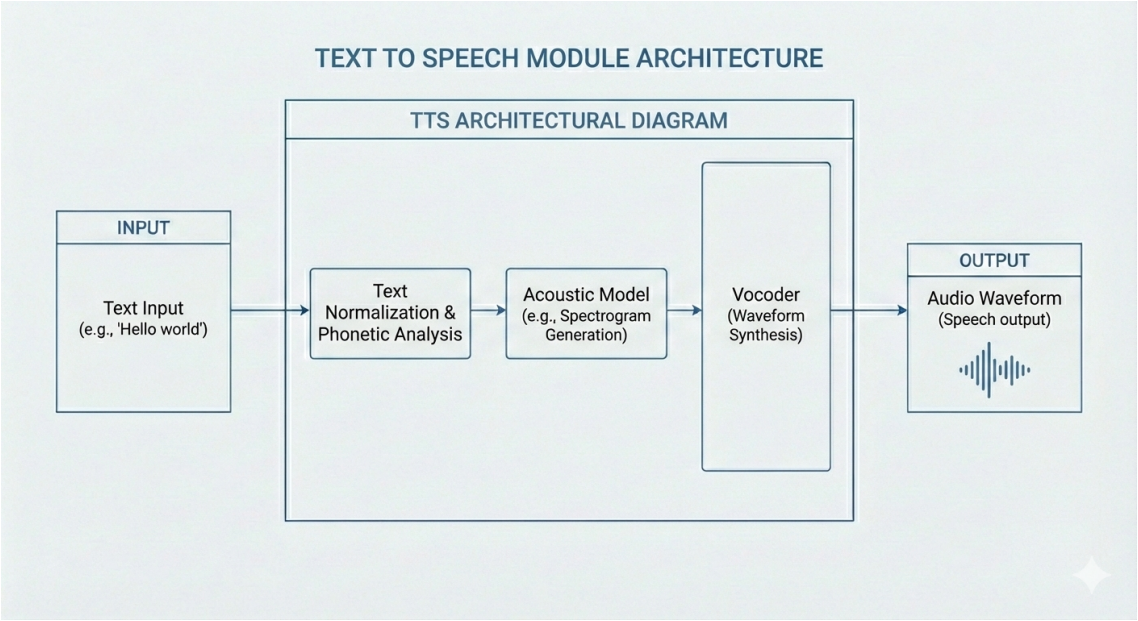

Figure 4.3: TTS Architecture

4.2.4 Language Understanding and Reasoning Module

The textual reasoning element is performed with an LLM-based language model, which is launched locally through a C++ inference server. The model manages the conversational context, understands queries of the user, and produces disease-related guidance in a way that is structured and interpretable. In the absence of an image, the language model gradually accumulates the information with the help of a question-based interaction policy. After collecting enough evidence, a possible disease is deduced, and the discussion changes into a disease-conscious mode. In cases where there is an image-based diagnosis, the language model preconditions all answers on the detected disease, which is important so that all statements in the interaction are consistent and relevant. The reasoning pipeline, as shown in Figure 4.4, follows a standard transformer-based workflow with the inputs initially being tokenized and embedded, and then executing stacked self-attention transformer blocks, then decoded by an output layer, and ultimately returns the resulting text response.

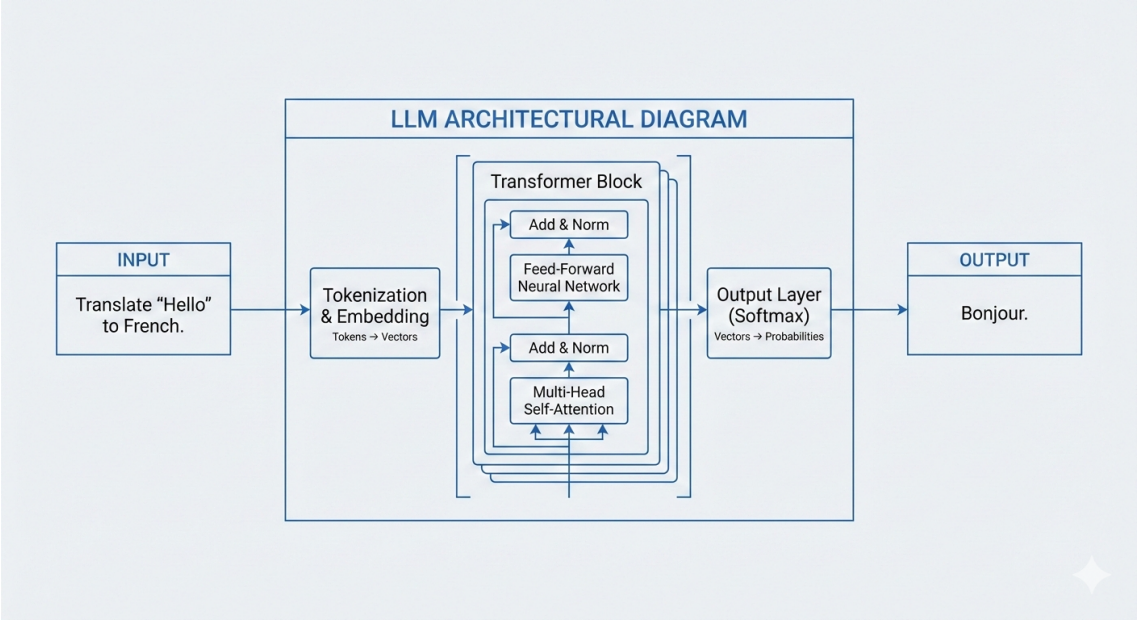

Figure 4.4: LLM Architecture

4.2.5 Combined System and Modular Workflow

The suggested system presented in Figure 4.5 is a modular and decoupled multimodal architecture, which is aimed at being flexible in interaction yet lightweight and capable of being implemented in resource-limited systems. The system incorporates the speech processing, visual analysis, and language reasoning modules, which are independent and intertwined with a clear workflow. One of the benefits of such a design is that it avoids the complex feature-level fusion of modalities, which would require a higher computational cost and would enhance the robustness of the system.

The system allows farmers to communicate with it through text, voice, and place images of plants, or a combination of any of the above. A Speech-to-Text (STT) module is then turned on when voice input is available, and then the text translation of the spoken query is produced. This enables the farmers who have limited literacy

skills or typing capability to use the system in their natural way of communication. In the case where the final response should be given in audio format, a Text-to-Speech (TTS) module is used to convert the text output it produces into audio. In the case when the farmer posts an image of a plant leaf, the CNN-based visual module is activated. This model has been trained on our personally curated dataset of plant disease images, so it can find the most probable disease in the uploaded photo. The CNN produces the predicted disease name (or class label), as opposed to visual features in high-dimensional form. The design option does not require any embedding of alignment or dimensional matching with the features of the language. The CNN-predicted disease name serves to provide a contextual cue, which is input to a locally deployed LLM-based reasoning model. In case no image is uploaded, the system will not utilize the visual module and will solely depend on the textual input in the form of direct text or STT output. This type of conditional activation scheme is efficient in the use of computational resources.

The LLM uses the name of the disease and the query of the farmer to retrieve information about the disease in our domain, agricultural information, a type of information stored in the agricultural domain, and is used to describe diseases, their symptoms, causes, preventive measures, and treatment guidelines. The language model can also create follow-up questions when they are needed (such as about symptoms, severity, or environmental conditions) to enhance confidence in diagnosis and user comprehension. Lastly, the LLM is able to produce a farmer-friendly and understandable response to the disease and measures taken. The output is presented in text form and can also be transformed into speech through the TTS module, and this completes the interaction loop.

Summing it up, the suggested system meets the main goal of this study in the sense that it offers a lightweight, offline-enabled, VQA-style agricultural diagnostic framework. The system allows identifying the disease and engaging in meaningful dialogue with the user with limited resources, which is only possible through an efficient CNN trained on a specialized dataset and a locally deployed LLM and speech interfaces. This modular solution proves the viability of implementing AI solutions, which are efficient in terms of resources, in agricultural decision support scenarios (especially in low-infrastructure-density and rural areas), where timely and reliable identification is necessary.

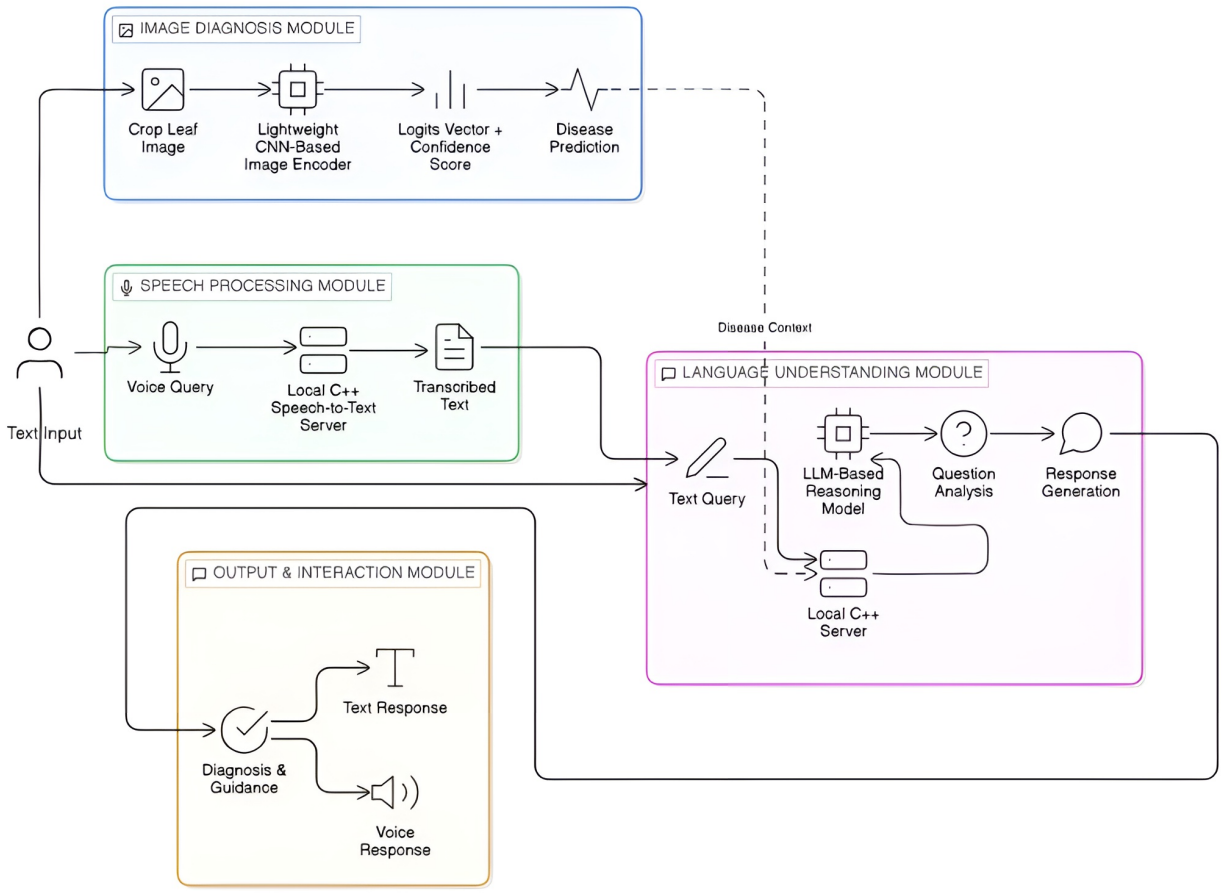

Figure 4.5: Combined System Architecture

4.3 Training Strategy and Hyperparameters Used for LLM and VLM Models

4.3.1 Training Strategy for Gemma 3-1B

The fine-tuning of Gemma 3–1B[43] was done with a supervised instruction-tuning (SFT) configuration designed for the agricultural diagnostic conversational context. The sample was split in 80% training and 20%testing and the Gemma-3 chat template was used to format all the samples before training. In order to make the training process possible on limited computational resources, parameter-efficient fine-tuning was done through LoRA with only selected language, attention, and MLP layers updated, and the base weights fixed. Besides this, the model was loaded in 4-bit quantized mode to minimize the use of memory. The complete settings of this experiment, including LoRA settings, train steps, batch size, learning rate, warmup, and optimizer, are in Table 4.1.

Table 4.1: Hyperparameters for Gemma 3 1B

Hyperparameter	**Value**
LoRA Rank (r)	8
LoRA Alpha	8
LoRA Dropout	0
Max_steps	30
Train Batch Size per Device	1
Learning Rate	2e-4
Logging Steps	1
Warmup Steps	5
Optimizer	adamw_8bit

4.3.2 Training Strategy for Gemma 3-4B

In this thesis, Gemma 3-4B was fine-tuned with text-only language modeling, in which the vision encoder was not used, and no image features were provided during training. This setup enabled the model to remain purely concentrated on learning the instruction response structure of our conversational dataset, and to regard every sample as a typical supervised fine-tuning (SFT) case. The training was carefully chosen to be done with a small budget of 30 steps to ensure consistency in all the model runs, and also in order to see rapid adaptation with limited computational resources. The batch size per device was chosen to be 1 to fit the 4B-parameter model within the scope of the GPU memory and allow maintaining stable parameter updates with the help of the trainer configuration. The rank $(r) = 16$ and alpha $= 16$ were used to apply LoRA to provide adequate adaptation capacity and minimize the number of trainable parameters with respect to the entire model. The dropout of LoRA was adjusted to 0, so that all update of the adapters would play their part to completion within the brief training timetable. The learning rate was 2×10^{-4} to facilitate effective learning with a few steps. Optimizing was done with fused AdamW (`adamw_torch_fused`), which is more efficient in training large models. To keep a close track of training, logging was performed at each step. Table 4.2 shows the entire hyperparameter setup of this Gemma 3–4B **ref42** run.

Table 4.2: Hyperparameters for Gemma 3 4B

Hyperparameter	**Value**
LoRA Rank (r)	16
LoRA Alpha	16
LoRA Dropout	0
Max_steps	30
Train Batch Size per Device	1
Learning Rate	2e-4
Logging Steps	1
Warmup Steps	— (not set)
Optimizer	adamw_torch_fused

4.3.3 Training Strategy for Ministral-3 3B

Supervised fine-tuning (SFT) was used by adjusting the model to multi-turn agricultural diagnostic dialogues based on the proposed data to fine-tune Ministral-3 3B[9]. The parameter-efficient LoRA was utilized to accomplish efficient domain adaptation with limited compute to achieve the relatively high capacity of adaptation, but maintain the weight of the base model unchanged. The fine-tuning procedure was executed with 30 steps per-device batches of 4, learning rate of 2e-4 with warmup stabilization. AdamW was used with 8- bit optimization to allow memory-efficient optimization, and the progress of training was tracked at every step. Table 4.3 shows all the hyperparameters that were applied to this experiment.

Table 4.3: Hyperparameters for Ministral 3 3B

Hyperparameter	**Value**
LoRA Rank (r)	32
LoRA Alpha	32
LoRA Dropout	0
Max_steps	30
Train Batch Size per Device	4
Learning Rate	2e-4
Logging Steps	1
Warmup Steps	5
Optimizer	adamw_8bit

4.3.4 Training Strategy for Qwen-3 1.7B

Qwen-3 1.7B[43] was optimized as a light conversational language model to be able to support farmer symptom descriptions and multi-turn agricultural question-answer conversations without any visual inputs. All training samples were framed as a guided conversation beginning with the text of the symptoms that were described by the farmer, then proceeding with guided follow-up questions and closing with professional recommendations and comments. Messages of the same role were concatenated to maintain the continuity of the conversation and minimize redundancy. A custom chat template was subsequently used to format the dataset in accordance with the instruction-tuning format, in order to have consistent role separation and response generation behavior. The supervised fine-tuning was used to train the model in 30 steps, with a focus on fast domain adaptation instead of complete retraining. The gradient accumulation with a batch size of 2 was chosen to achieve a compromise between the stability of the training and the memory efficiency on low-resource hardware. The 4-bit loading was quantized to allow effective fine-tuning whilst maintaining a high quality of language, and 32 was selected as the quantized rank and alpha of the model to provide enough room to adapt the model to the needs of the agricultural domain with limited data. LoRA dropout was initialized to zero to store as much task-specific information as possible when training in short runs. We used a learning rate of 2e-4 to get rapid convergence given a limited number of training steps, and AdamW with 8-bit optimization to minimize memory usage.

Table 4.4 contains all the hyperparameters of this training set.

Table 4.4: Hyperparameters for Qwen 3 1.7B

Hyperparameter	**Value**
LoRA Rank (r)	32
LoRA Alpha	32
LoRA Dropout	0
Max_steps	30
Train Batch Size per Device	2
Learning Rate	2e-4
Logging Steps	1
Warmup Steps	5
Optimizer	adamw_8bit

4.3.5 Training Strategy for Ministral-3 3B Instruct (VLM)

Ministral-3 3B[9] Instruct was tested in a multimodal vision-language configuration to produce disease-related responses of paired leaf images and queries by a farmer. This model was trained with a small per-device batch size and gradient accumulation to fine-tune on a supervised fine-tuning of 30 steps with small computational and storage requirements. In contrast to other multimodal experiments, the adaptation of the LoRA was not used in the case of such a configuration; thus, the parameters of LoRA were not used in this run. AdamW was used to optimize the model with 8-bit optimization to minimize memory usage and make the process of updating the parameters efficient. The entire hyperparameter settings of this fine-tuning configuration are summed up in Table 4.5.

Table 4.5: Hyperparameters for VLM Ministral 3 3B Instruct

Hyperparameter	**Value**
LoRA Rank (r)	— (no LoRA used)
LoRA Alpha	—
LoRA Dropout	—
Max_steps	30
Train Batch Size per Device	2
Learning Rate	2e-4
Logging Steps	1
Warmup Steps	5
Optimizer	adamw_8bit

4.3.6 Training Strategy for Qwen-3 VL 4B

Qwen-3 VL 4B[43] has been optimized as a vision-language model (VLM) to concurrently engage the images of plants and textual or spoken queries provided by

the farmer in a unified conversational paradigm. The supervised fine-tuning was performed on 30 steps to train with the goal of efficient domain adaptation, not necessarily efficient optimization. The gradient accumulation version of a per-device batch size of 2 was picked to ensure that training stability is maintained, given the memory constraint. An Unsloth vision datacollator was used so that image-text pairs were handled correctly and multimodal structure was maintained during training. To trade off expressive adaptation capacity and computational efficiency, a LoRA rank and alpha of 16 were selected, which is of particular concern to larger multimodal models. LoRA dropout was fixed at zero to retain all the learned task-specific representations throughout the short training schedule. The 2e-4 learning rate allowed the model to converge quickly with fewer steps, whereas AdamW with 8-bit optimization was able to save memory without jeopardizing the stability of the optimization. Table 4.6 gives the final hyperparameter setup of this model.

Table 4.6: Hyperparameters for VLM Qwen3-VL 4B

Hyperparameter	**Value**
LoRA Rank (r)	16
LoRA Alpha	16
LoRA Dropout	0
Max_steps	30
Train Batch Size per Device	2
Learning Rate	2e-4
Logging Steps	1
Warmup Steps	5
Optimizer	adamw_8bit

4.4 Fine-tuning Existing Models

In our research, we did fine-tuning on several pre-trained foundation models with our custom conversational dataset. Every sample was organized into a multi-turn dialogue: the user initiates the discussion with the description of the symptoms (`symptom_text`), a guided question-answer conversation (`qa_flow`) and finishes with the final response of the assistant with advice and notes. In the case of the multimodal (vision-language) configuration, every conversation was further accompanied by a corresponding leaf disease image sampled in a labeled image folder hierarchy. We trained Unsloth in SFTTrainer with TRL to allow the models to be fine-tuned with limited compute and to use LoRA to fine-tune the models. Where available, quantized loading (4-bit) was employed, whereas some Ministral executions were performed without 4-bit quantization to remain compatible. We used identical hyperparameters in all experiments, namely max steps = 30, learning rate = 2×10^{-4}, and gradient accumulation to approximate large batch sizes. The training was done in a single GPU setting.

4.4.1 Gemma-3 1B Model

Figure 4.6 shows the training loss curve of the Gemma-3 1B[43] model in 30 steps. The loss exhibits a significant overall downward pattern and is decreasing in value starting at a higher point towards a lower and more stable value as the training advances, and this is a sign of successful learning and convergence. Despite some minor changes in the short-run and minor spikes at the intermediate stages, the overall tendency is steadily declining, indicating that the optimization procedure is stable, and the model is gradually becoming more and more aligned with the training data.

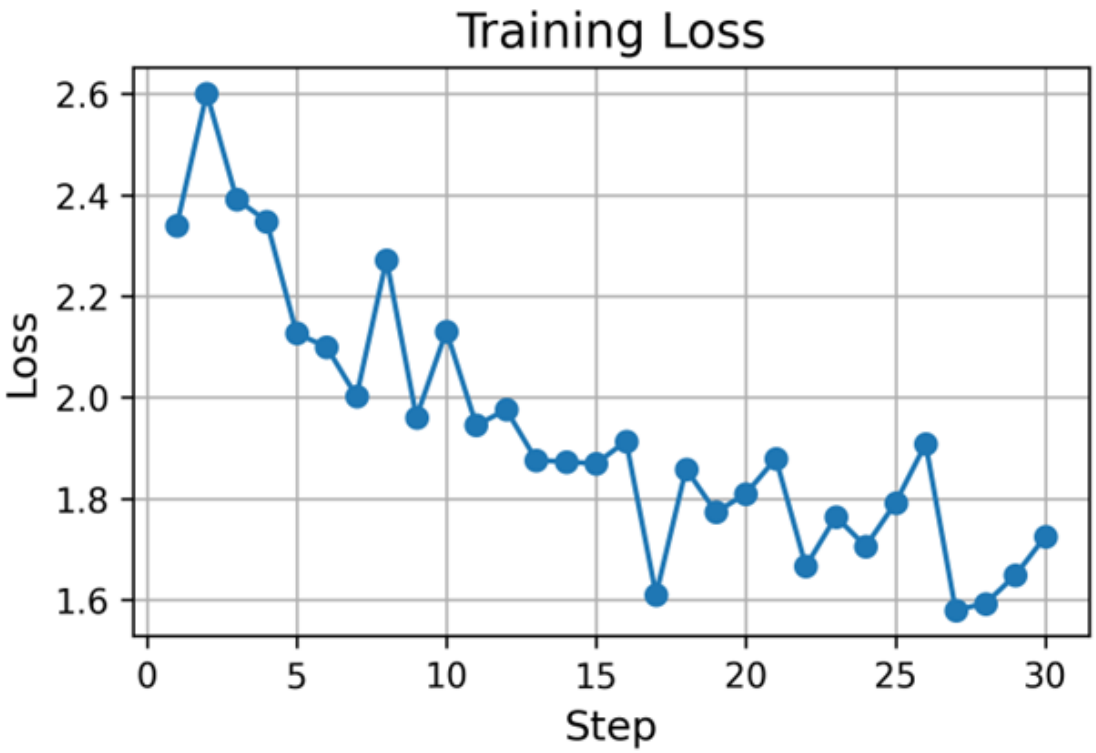

Figure 4.6: Gemma-3 1B Training Loss Curve.

4.4.2 Gemma-3 4B Model

Figure 4.7 represents the training loss curve of the Gemma-3 4B[43] model in 30 steps. Loss becomes smaller in the initial steps, which means that fine-tuning learns very fast in the initial stages, and then tends to stabilize at a lower value. The curve remains fairly consistent at the point after which there are small deviations and several small spikes, implying that the model is approaching the point of convergence but continues to achieve minor improvements.

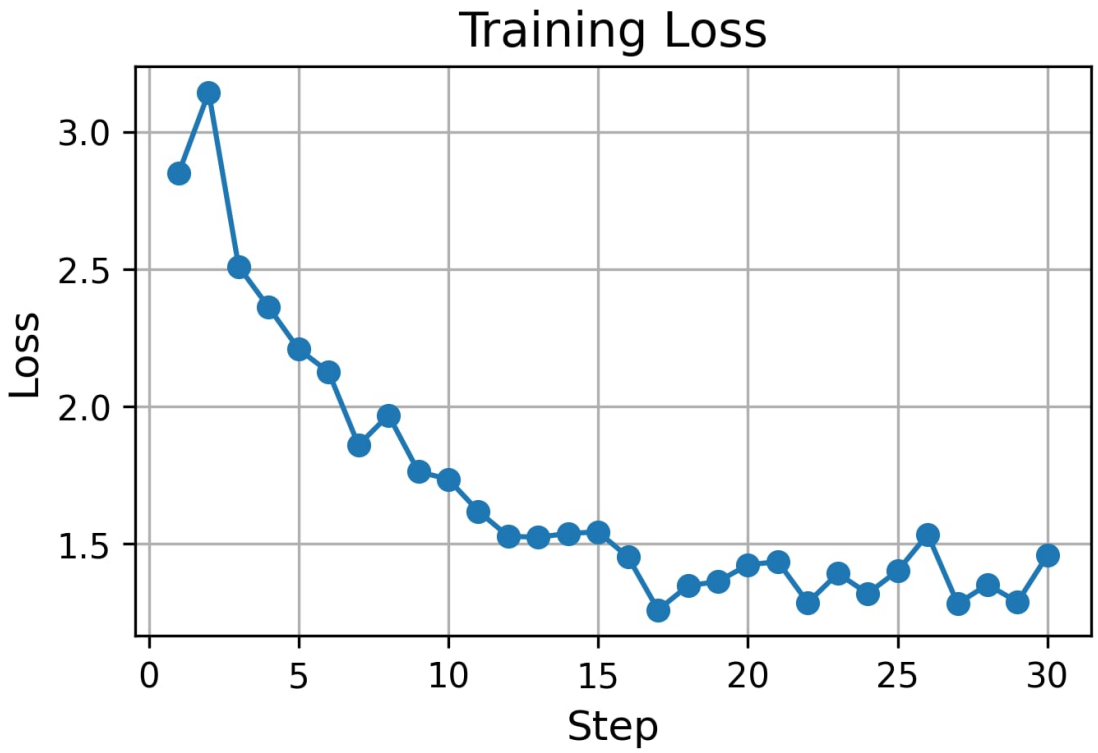

Figure 4.7: Gemma-3 4B Training Loss Curve.

4.4.3 Qwen3 1.7B Model

The loss curve of the Qwen3 1.7B[44] model in 30 steps is drawn in Figure 4.8. The loss starts to decline at the initial stages with a rapid initial learning, and then starts to slow down at later stages of training. The curve takes a comparatively stable course but with few fluctuations after the midpoint, meaning that the model is nearing convergence and the optimization process is stable as it continues to improve its performance.

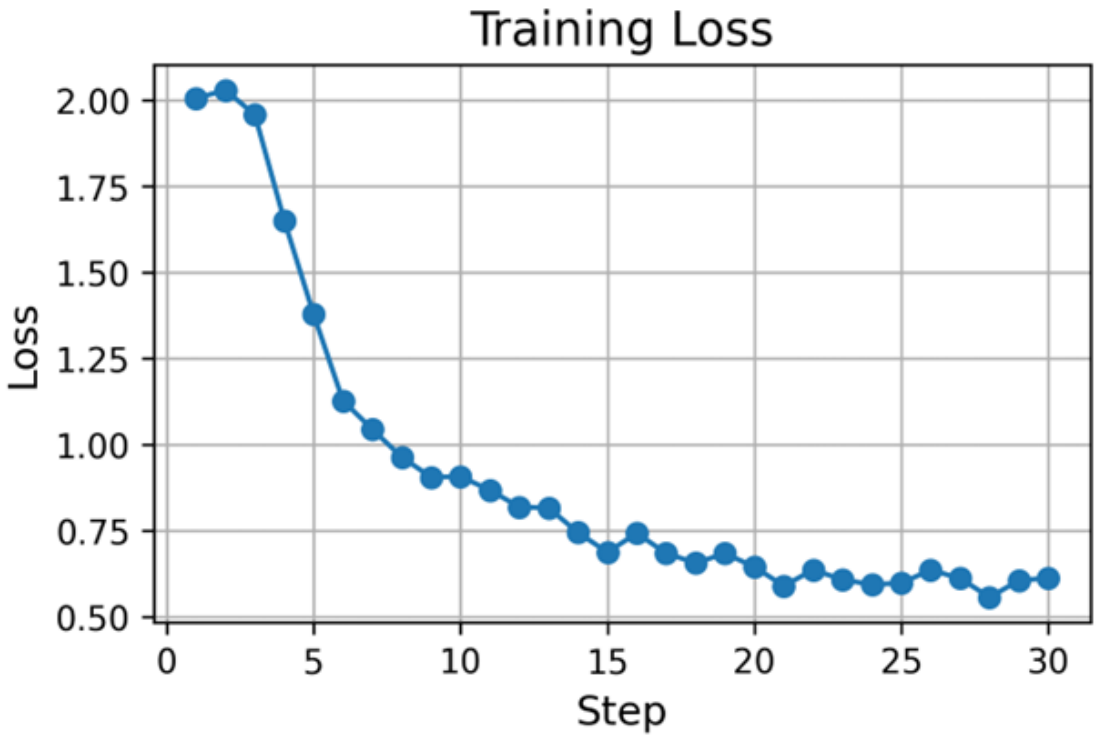

Figure 4.8: Qwen3 1.7B Training Loss Curve.

4.4.4 Ministral-3 3B Model

The training loss curve of Ministral-3 3B[9] model is presented in Figure 4.9, and it was calculated in 30 steps. The loss decreases rapidly during the initial steps, denoting rapid early learning followed by a decrease at a slower rate as the training advances. Beyond the mid-training point, the curve levels off to minor fluctuations, but with occasional small spikes indicating that the model is approaching convergence and still making marginal increases.

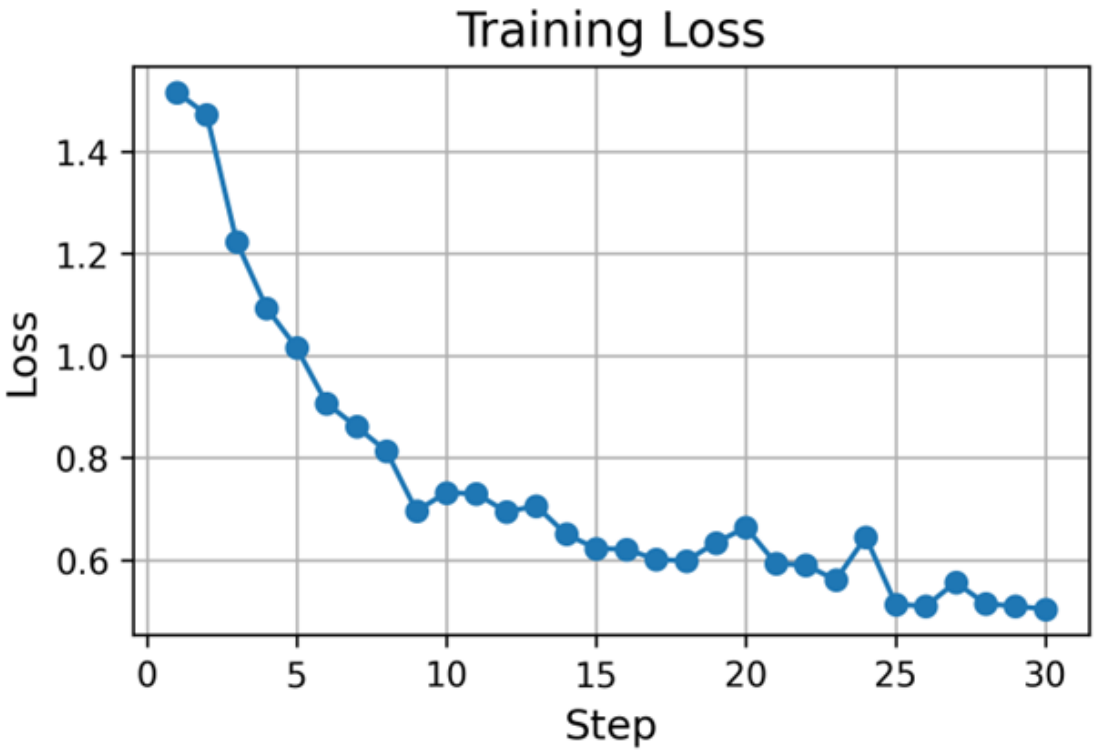

Figure 4.9: Ministral-3 3B Training Loss Curve.

4.4.5 Qwen3-VL 4B Model (VLM)

In Figure 4.10, the training loss curve of Qwen3-VL 4B[44] model (VLM) is shown in 30 steps. The loss progressively decreases at the initial stages, although more steeply in the first stages, which represents rapid learning and adaptation to the multimodal training goal. Following the initial drop, the curve continues to decrease at a slower rate with only slight variations, and this is a sign of a stable optimization and a smooth convergence of the model to the latter steps of the performance.

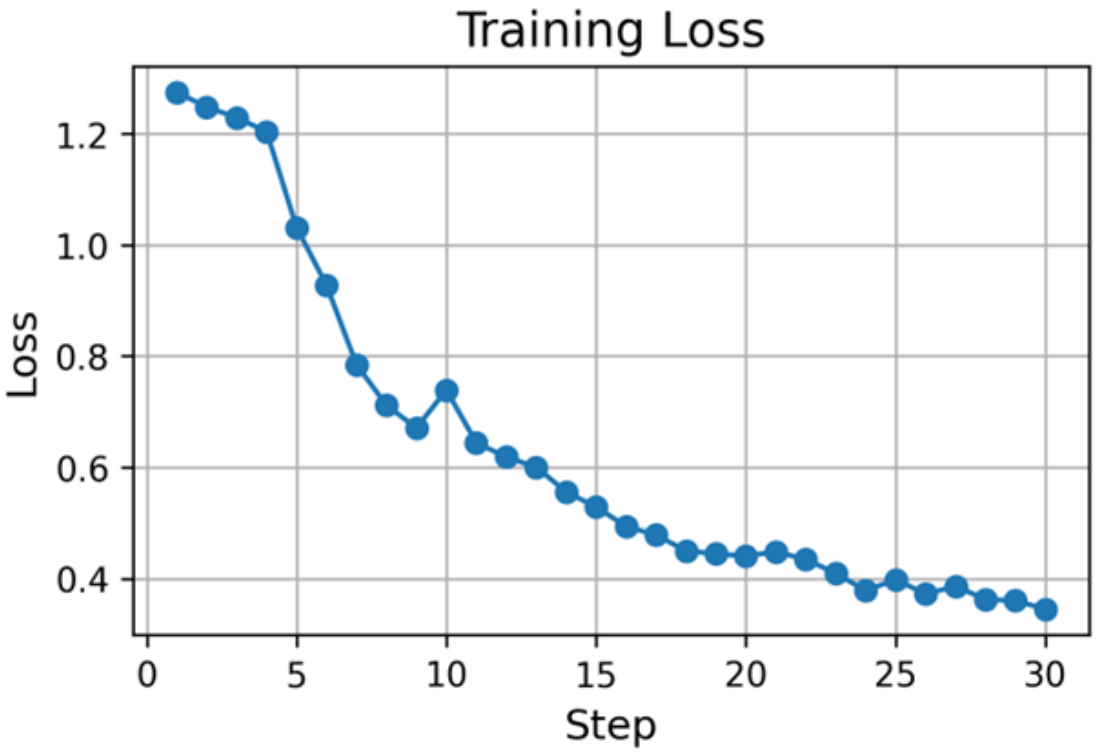

Figure 4.10: Qwen3-VL 4B Training Loss Curve.

4.4.6 Ministral-3 3B Instruct (VLM)

Figure 4.11 demonstrates the training loss curve of 30 steps of the Ministral-3 3B[9] Instruct (VLM) model. The loss begins with a high value and decreases very quickly in the first few steps, which refers to a quick learning process and good early adaptation. Once the training passes approximately the mid-point, the curve goes flat and stays well under a very low level with only slight deviation, implying that the convergence is steady and that the model has become highly optimized in its fit under the current training conditions.

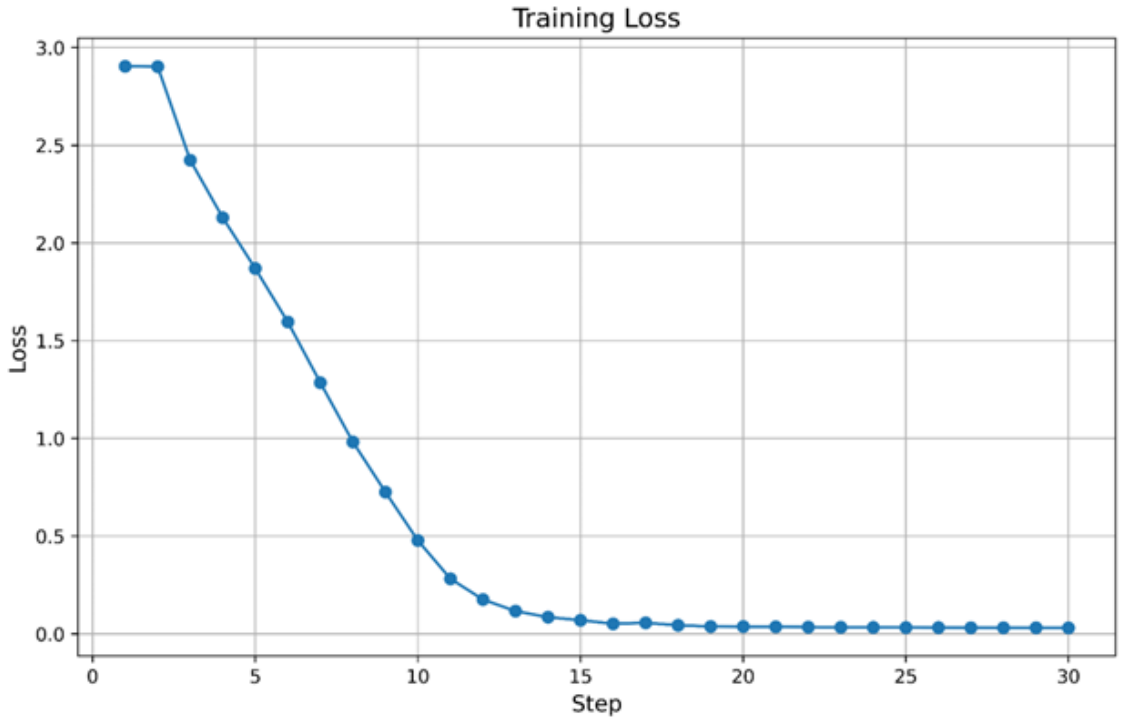

Figure 4.11: Ministral-3 3B (VLM) Training Loss Curve.

4.4.7 Image Classification Models

To measure how effective lightweight image encoders are in the classification of rice diseases under the conditions of resource scarcity, three representative deep learning architectures were chosen: EfficientNet-B0, MobileNetV3-Small, and MobileViT-XS. These models represent three different design paradigms, such as compound-scaled convolutional networks, mobile-optimized CNNs, and hybrid CNN-Transformer models, which makes them ideal to compare fairly and informatively, in an agricultural diagnostic context.

EfficientNet-B0 was chosen because of its compound scaling policy that balances the depth, width, and input resolution of the network to optimize the accuracy at a moderate computation cost. The training and test accuracy improve gradually with an increase in epochs, as indicated in Figure 4.12, which implies unchanging learning and effective generalization. In accordance with this, Figure 4.13 indicates a definite negative trend in training and test loss, indicating convergence and decreased error with time. When combined, Figures 4.12 and 4.13 show that EfficientNet-B0 is a good performance benchmark in the task of classifying rice diseases when resources are limited.

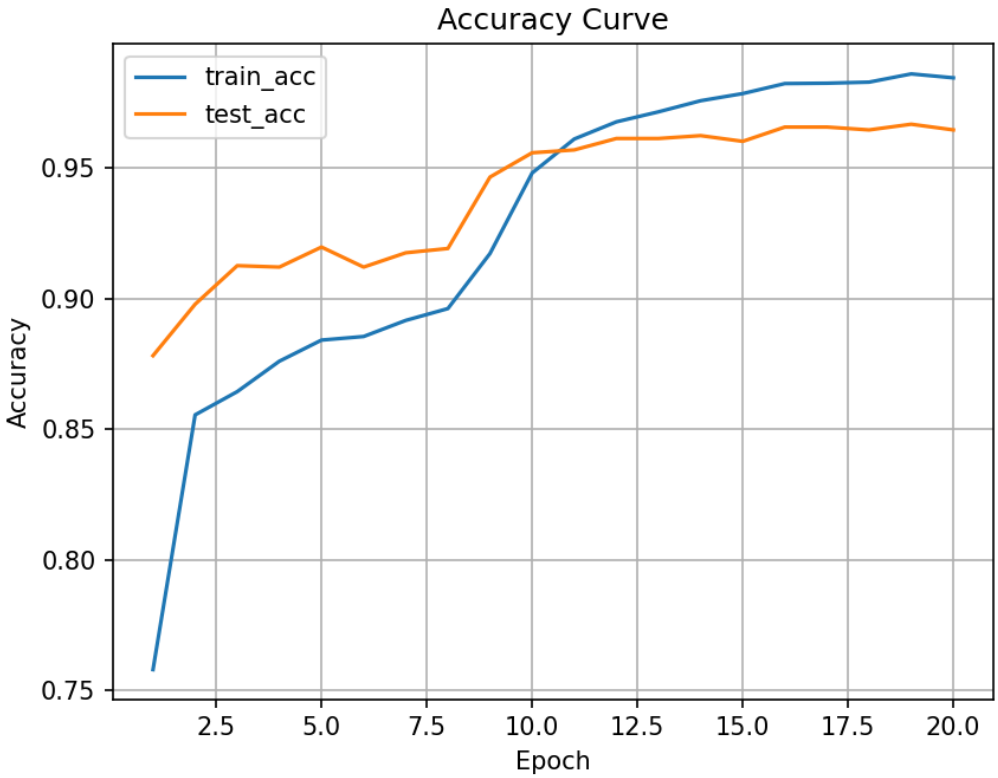

Figure 4.12: EfficientNet-B0 Training and Test Accuracy Curve

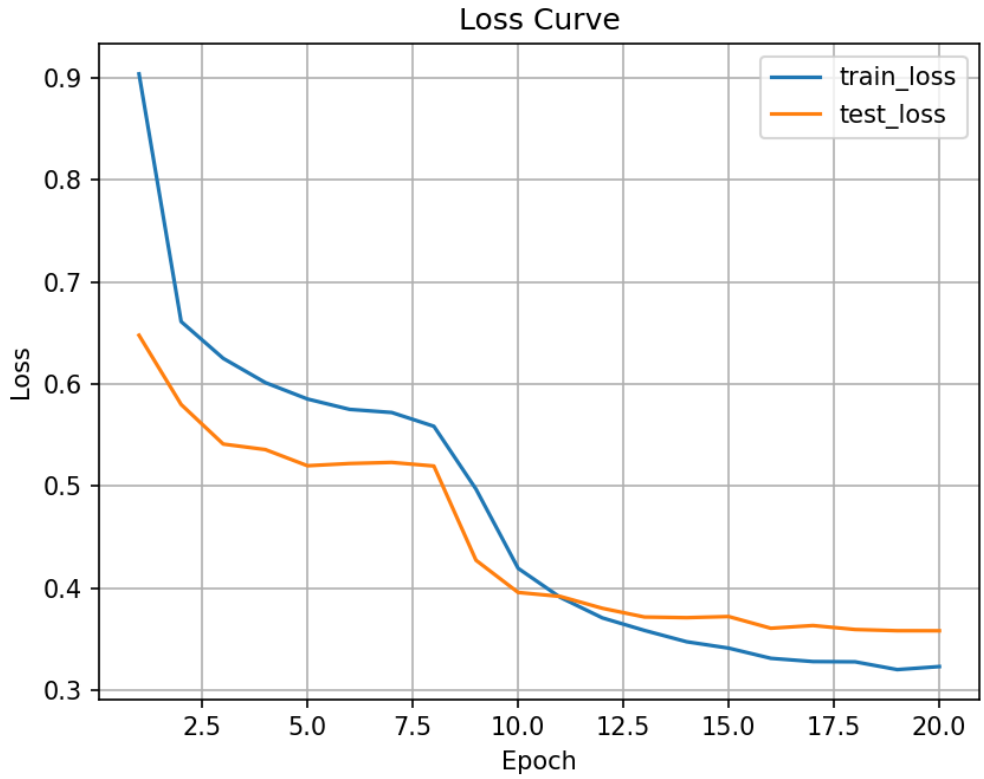

Figure 4.13: EfficientNet-B0 Training and Test Loss Curve

MobileNetV3-Small is a mobile-friendly model that has been optimized to run on low power and at the edge. It consumes less memory and inference cost through depthwise separable convolutions, inverted residual blocks, and lightweight attention mechanisms that do not compromise performance in classification. Training behavior is shown in Figure 4.14, in which the training and test accuracy increase rapidly and level off; the rapid convergence is typical of practical mobile training and fine-tuning. On the same note, Figure 4.15 indicates that the training and test loss reduce steadily, which indicates the behavior of stable optimization and controlled generalization. All in all, Figures 4.14 and 4.15 prove that MobileNetV3-Small can be deployed to a low-budget smartphone that is widely used in a rural setting.

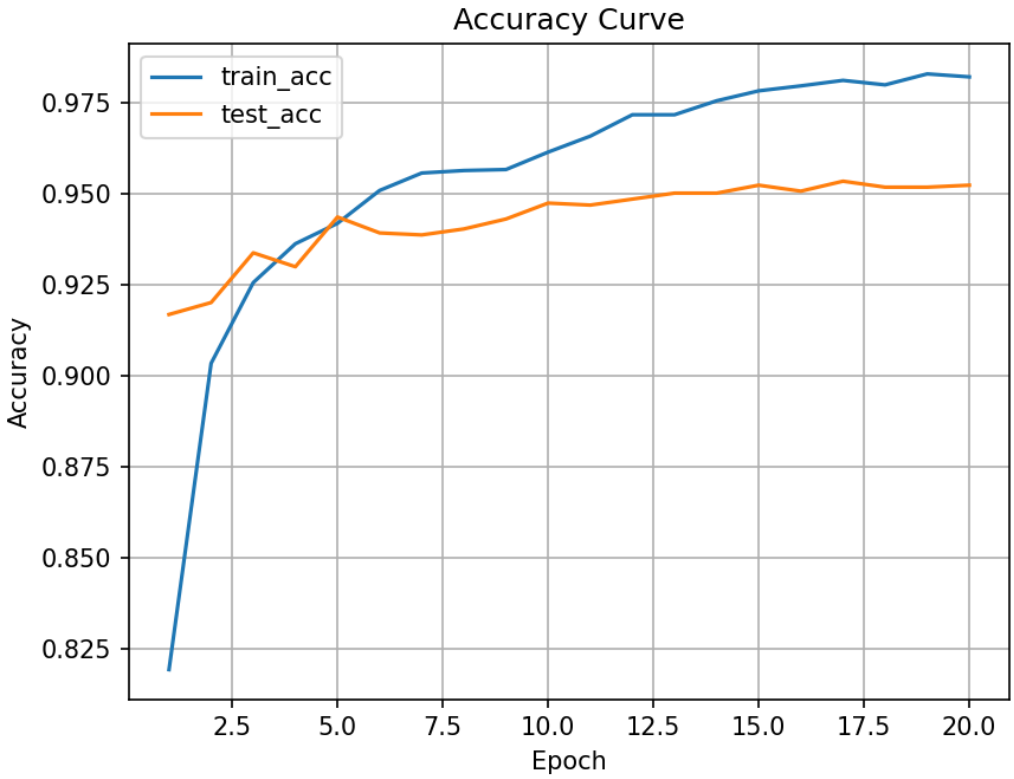

Figure 4.14: MobileNetV3-Small Training and Test Accuracy Curve

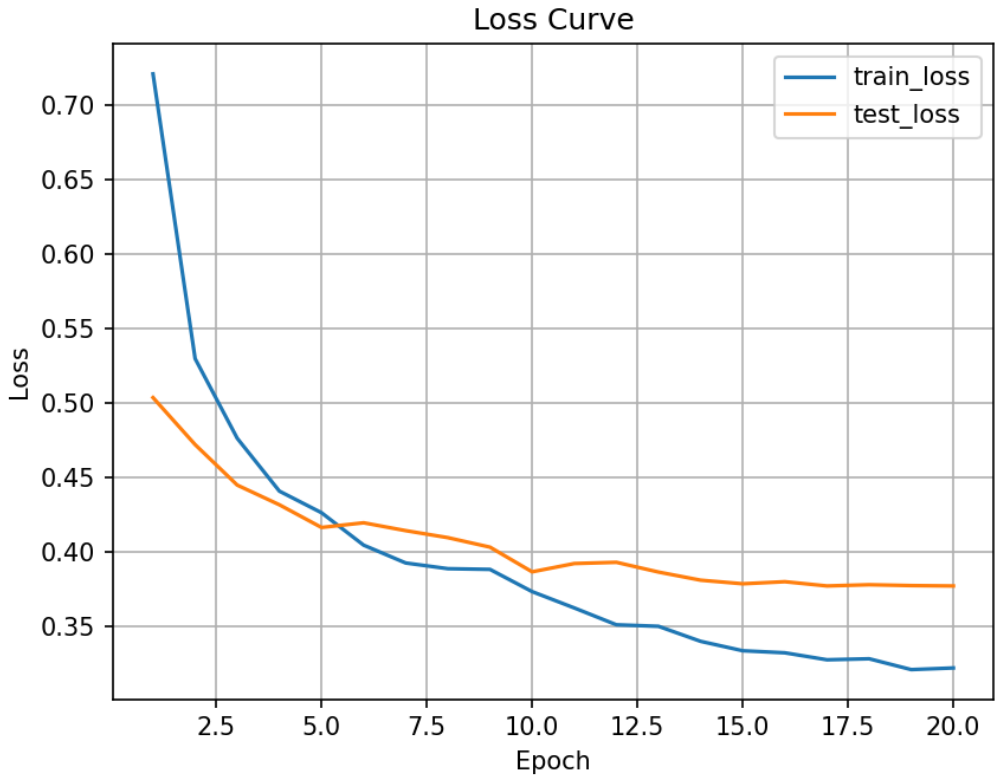

Figure 4.15: MobileNetV3-Small Training and Test Loss Curve

MobileViT-XS is a hybrid architecture that combines the convolutional layers with transformer-based self-attention mechanisms. The convolutional components refer to the local spatial features, whereas the transformer blocks follow the long-term dependencies as well as the long-term higher context, allowing better representation within the lightweight constraints. The accuracy trend has been shown in Figure 4.16, whereby the training curve shows great performance, and the test accuracies are constant in each epoch, meaning that generalization is consistent. The loss behavior is represented in Figure 4.17, and this reflects a smooth convergence and minimized error during training. Combined, Figures 4.16 and 4.17 point to the fact that MobileViT-XS has strong learning dynamics and stable generalization, which makes it a solid lightweight solution to the rice disease classification.

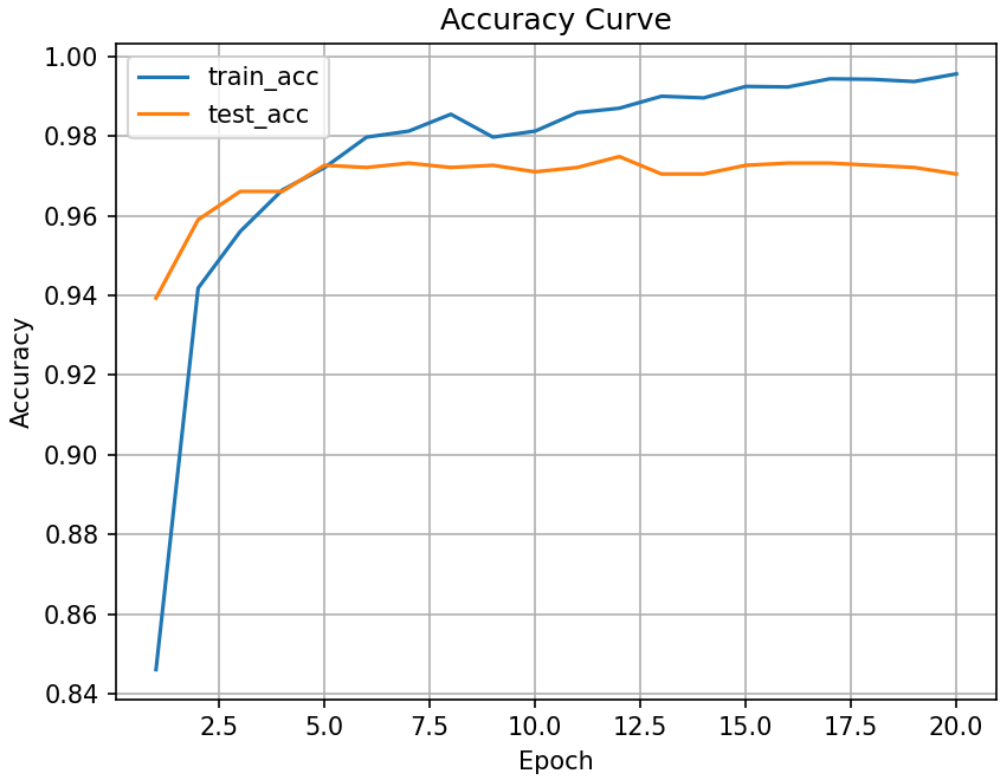

Figure 4.16: MobileViT-XS Training and Test Accuracy Curve

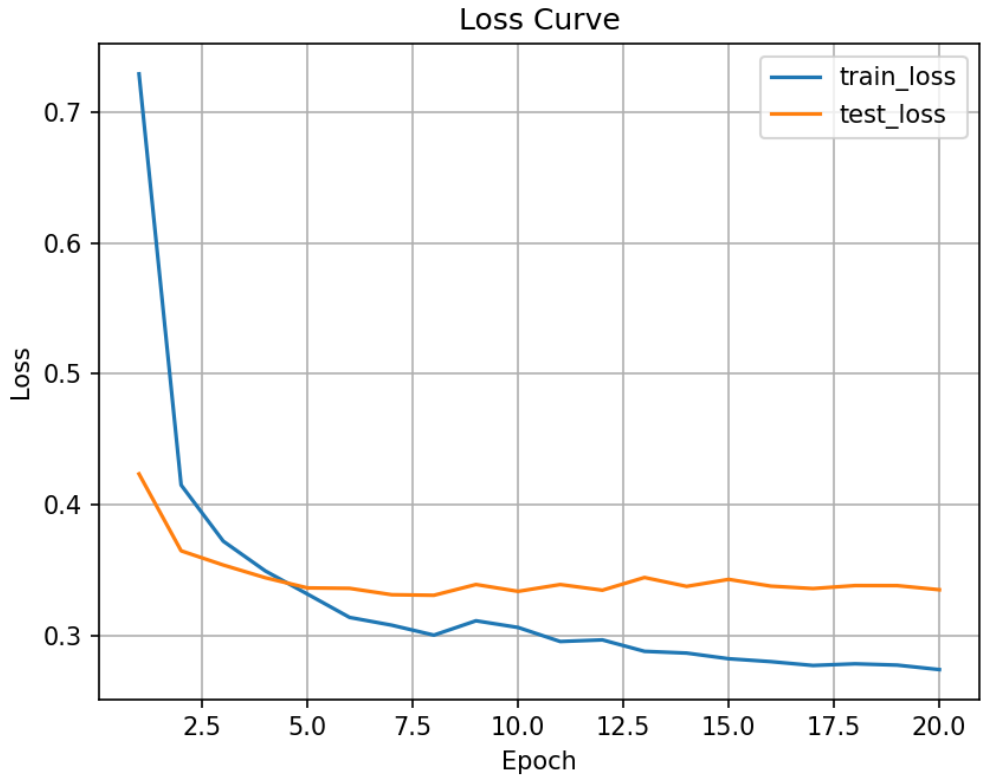

Figure 4.17: MobileViT-XS Training and Test Loss Curve

In Figures 4.16 and 4.17, it can be seen that the accuracy and loss curves of MobileViT-XS are superior in terms of classification performance and convergence during training and fine-tuning phases.

The obtained test accuracies of the three tested models are summed up as follows:

- MobileViT-XS: approximately 97.49%
- EfficientNet-B0: approximately 96.67%
- MobileNetV3-Small: approximately 95.35%

MobileViT-XS was the most accurate in classification compared to the other two architectures. This observation is not surprising, since with the transformer-based self-attention mechanism, MobileViT can be used to model the global spatial relationships on the entire leaf surface. This global context modeling is found to be especially beneficial in the classification of rice disease, where the symptoms are usually in the form of spatially distributed patterns as opposed to localized features.

4.4.8 Comparative Evaluation of Fine-Tuned Bangla STT Models

This research compares five speech-to-text (STT) models in Bangali in an offline and CPU-constrained environment: Whisper-Tiny, Whisper-Base, Whisper-Small,

Whisper Large, and facebook/wav2vec2-large-xlsr-53. The objective of this comparison is to decide on a model that makes a reasonable tradeoff between (i) quality of transcription, (ii) inference-latency, and (iii) storage/runtime cost, to be used offline in mobile. All Whisper variants follow an encoder–decoder Transformer design, where an audio encoder produces latent acoustic representations and a text decoder generates the transcript autoregressively. By contrast, wav2vec2-large-xlsr-53 is a self-supervised Transformer encoder model, which is trained to learn to represent speech; in an ASR pipeline, it generates token probabilities, such as CTC decoding, to create transcriptions. These architectural differences are likely to produce differences in the behavior of CPU accuracy and latency.
All the models were trained using speech recognition on Bangla speech recognition with a larger Bangla audio corpus to be more adaptable to the language, capture the variation of the pronunciation, and cope with domain vocabulary.

Baseline Model Characteristics

In order to determine an accurate resource baseline of the Whisper family and Wav2vec2, the approximate storage size and optimal reported Word Error Rate (WER) are as follows (smaller is better):

Table 4.7: Baseline model characteristics of the evaluated STT models

Model	**Approx. Size**	**WER**	**CER**
Whisper-Tiny	100–200 MB	0.941	0.4828
Whisper-Base	200–300 MB	0.353	0.092
Whisper-Small	∼1 GB	0.2353	0.0575
Whisper-Large	∼3–4 GB	0.176	0.046
wav2vec2-large-xlsr-53	∼1.27 GB	0.8113	0.3333

Table 4.7 summarises the baseline storage footprint, WER, and CER of the models of the evaluated STTs. This baseline means that larger variants of Whisper tend to decrease WER and CER but raise storage footprint significantly, which directly impacts the mobile feasibility.

Chapter 5

Result Analysis

5.1 Performance Evaluation

The performance of the proposed language reasoning module was measured by a wide range of lexical, semantic, robustness, and fluency-based measurements. In natural language generation and question-answering, it is not enough to use a single metric, where answers can be correct in semantics but have different lexicons compared to answers in the reference. Thus, several complementary evaluation measures were used to make a valid and thorough measure of the model performance, particularly when it comes to agricultural decision support.

5.1.1 Evaluation Criteria

- **BLEU (Bilingual Evaluation Understudy):** BLEU measures the n-gram overlap between the generated response and the reference answer. It is commonly used to assess surface-level similarity in text generation tasks. However, BLEU alone may underestimate performance when valid responses use different wording.

$$\text{BLEU} = \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{5.1}$$

  where p_n represents the modified n-gram precision and w_n denotes the weight assigned to each n-gram.

- **METEOR (Metric for Evaluation of Translation with Explicit Ordering):** METEOR improves upon BLEU by incorporating stemming, synonym matching, and recall, making it more suitable for evaluating semantically equivalent but lexically varied responses.

$$\text{METEOR} = \frac{10 \times P \times R}{R + 9P} \tag{5.2}$$

  where P is precision and R is recall computed over aligned unigrams.

- **ROUGE (Recall-Oriented Understudy for Gisting Evaluation):** ROUGE evaluates the overlap between generated and reference texts, with a stronger

emphasis on recall. In this work, ROUGE-1, ROUGE-2, and ROUGE-L were used.

$$\text{ROUGE-N} = \frac{\sum_{n\text{-gram} \in \text{Ref}} \min(\text{Count}_{\text{gen}}, \text{Count}_{\text{ref}})}{\sum_{n\text{-gram} \in \text{Ref}} \text{Count}_{\text{ref}}} \tag{5.3}$$

ROUGE-L is based on the Longest Common Subsequence (LCS), capturing sentence-level structure similarity.

- **BERTScore:** BERTScore evaluates semantic similarity using contextual embeddings from pretrained transformer models. Unlike n-gram metrics, it captures meaning rather than exact word matches. Precision, Recall, and F1-score were computed.

$$\text{BERTScore}_{F1} = \frac{2 \times P \times R}{P + R} \tag{5.4}$$

  This metric is particularly effective for domain-specific question answering where multiple correct formulations may exist.

- **Cosine Similarity:** Cosine similarity measures the semantic closeness between sentence embeddings of the generated and reference responses.

$$\text{Cosine Similarity} = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \tag{5.5}$$

  Higher values indicate stronger semantic alignment between responses.

- **Word Error Rate (WER):** WER evaluates the robustness of generated responses to transcription noise, which is especially important for voice-based interaction using Speech-to-Text (STT).

$$\text{WER} = \frac{S + D + I}{N} \tag{5.6}$$

  where S is substitutions, D is deletions, I is insertions, and N is the number of words in the reference text.

- **Perplexity:** Perplexity measures the fluency and confidence of the language model by estimating how well it predicts the next word in a sequence. Lower perplexity indicates more fluent and coherent text generation.

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N} \log P(w_i)\right) \tag{5.7}$$

LLM-as-Judge Conversation Score: Meaning, Calculation, and Interpretation

LLM-as-Judge (also known as model-based evaluation) is a type of evaluation in which a language model is employed to evaluate the outputs of a second language model. Rather than scoring text with a single reference answer and measuring the overlap (like BLEU/ROUGE), the judge model reads the conversation (or response) and a rubric prompt specifying what the term good means (e.g., correctness, completeness, relevance, safety, tone), and then scores or rates based on that rubric. This

is a popular component of more recent LLM evaluation pipelines since it has the ability to capture qualitative elements that surface-form metrics do not. An average LLM-as-Judge system results in one scalar score on each conversation. Conceptually, in the case of every conversation (c_i) and every rubric prompt (P), a judge model (J) will give a score:

$$s_i = J(P, c_i), \quad s_i \in [0, 1]$$

The most often used normalization is the "0–1" range since it is easy to compare the scores across the runs and dashboards and provides easy statistical summaries. This pattern is used by different evaluation libraries and platforms, but with various rubrics and styles of prompting; however, the general concept is the same: the judge converts qualitative judgment into a numeric signal.
Once a list of scores ($S = \{s_1, s_2, \ldots, s_n\}$) across ($n$) conversations is available, the "conversation score (statistical summary)" is computed as descriptive statistics over that list. The mean is the average score ($\mu = \frac{1}{n}\sum_{i=1}^{n} s_i$). The median is the 50th percentile $Q(0.50)$, meaning half of the conversations score below it and half above. Percentiles like $p25/p75$ are the 25th/75th quantiles $Q(0.25)$, $Q(0.75)$, and $p95$ is the 95th quantile $Q(0.95)$. In many implementations, these quantiles are computed using standard library functions (for example, pandas `Series.quantile(q)` for $q \in [0, 1]$). Min and max are just the smallest and the largest individual scores of conversations that are witnessed. Combined, mean/median/percentiles/min/max depict the consistency or variability of performance: the center (mean/median), the dispersion (p25–p75), the tails (p95 and min), and the extremes (min/max). These statistics do not formulate new judgments; they summarize the scoring of the judge of the batch.
Score scale:
A numeric score is defined in ([0, 1]), where higher means "better" under the rubric. A practical interpretation commonly used with 0–1 judge scores:

- 0.00 – 0.20: Fails objective; unusable / largely incorrect
- 0.20 – 0.40: Weak; partial completion with major issues
- 0.40 – 0.60: Okay; mixed quality, noticeable gaps/errors
- 0.60 – 0.80: Good; mostly correct/helpful with minor issues
- 0.80 – 0.90: Very good; strong completion, small nitpicks
- 0.90 – 1.00: Excellent; near-ideal per rubric

5.1.2 Motivation for Metric Selection

The inclusion of n-gram-based metrics (BLEU, METEOR, ROUGE), semantic similarity metrics (BERTScore, Cosine Similarity), robustness assessment (WER), and fluency assessment (Perplexity) makes sure to balance and appropriate assessment framework for the task. Such a multi-metric method is needed in the agricultural question-answering system, in which semantic accuracy, clarity, and robustness to speech input are more critical than exact lexical matching.

5.2 Analysis of Designed Solution

The comparison of BLEU scores shows that Gemma 3-1B[42] has the highest score in the set of tested models, which could be taken as a sign of the improved correspondence of generated and reference answers. This is compared in Figure 5.1. Gemma 3-4B[43] is also followed by an average decrease, whereas both Qwen 3-1.7B[44] and Qwen 3-4B[43] have significantly lower values of BLEU, which is lower quality of generation in the same environment. Among the Ministral versions, Ministral-3-3B-Instruct has the lowest score on the BLEU scale, whilst Ministral-3-3B exhibits a greater BLEU score than the Instruct version, which is a positive indication of better quality output. The general findings indicate that Gemma-based models would be more applicable in the assessed language generation case.

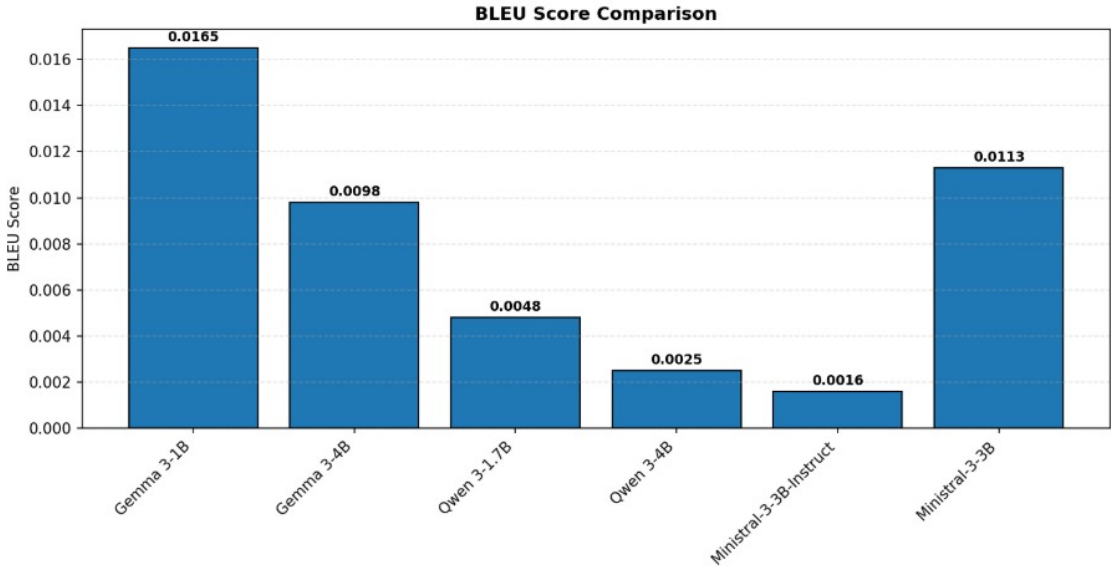

Figure 5.1: BLEU Score Comparison Across Language Models

Subsequently, after comparing the METEOR scores of models evaluated for language, METEOR is the evaluation metric in which the higher the score, the better the quality of text generation/translation. Gemma 3-1B[43] and Gemma 3-4B[36] are the two best models (0.1095) and (0.0780), respectively. The performance of both models of Qwen is moderate and very similar (around 0.056), and Ministral-3-3B-Instruct has the lowest performance (0.0367). Nonetheless, Ministral-3-3B scores better (0.0968), which is closer to the Gemma models than the Instruct one. This is compared in Figure 5.2.

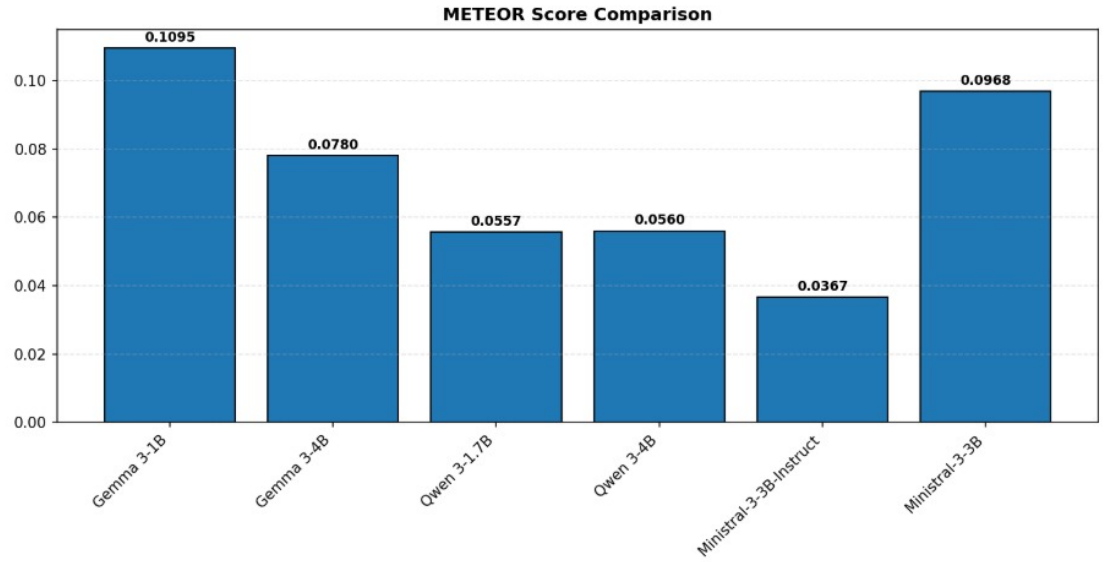

Figure 5.2: METEOR Score Comparison Across Language Models

Figure 5.3 compares the ROUGE-1, ROUGE-2, and ROUGE-L scores of six language models to test their quality in summarizing and text generation. Gemma 3-1B[43] has always scored the best on all three measures, showing higher unigram overlap, bigram matching, and sequence-level similarity with the reference texts. The second-best model is Gemma 3-4B[36] with a small but good score. Qwen 3-1.7B[43] shows fair performance and is still higher than Qwen 3-4B[43] in measures. In the case of the Ministral variants, Ministral-3-3B-Instruct[9] has the lowest ROUGE scores, and Ministral-3-3B[9] has significantly higher ROUGE values, implying that Ministral-3-3B has better content overlap and coherence than Instruct.

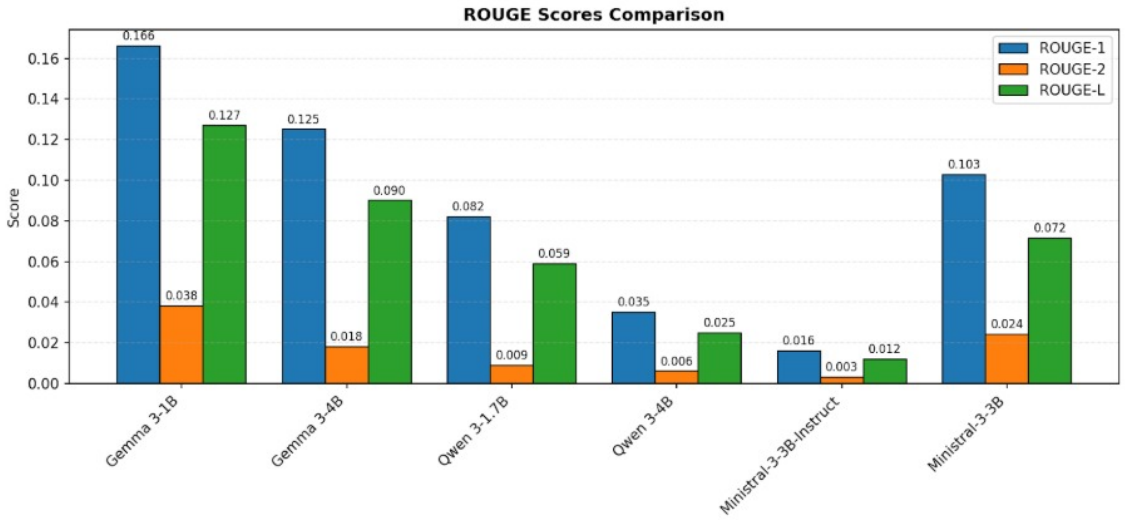

Figure 5.3: ROUGE Score Comparison Across Language Models (ROUGE-1, ROUGE-2, ROUGE-L)

WER (Word Error Rate) can be defined as the number of word-level errors a speech-to-text model can make relative to the actual transcript, in comparison to a lower WER value indicating improved transcription. Based on the chart in Figure 5.4,

we can see that Gemma 3-4B[43] (1.24) and Gemma 3-1B[43] (1.44) work best with extremely low error rates, and Qwen 3-1.7B[44] (1.65) also works well. Ministral-3-3B[9] has moderate WER (2.16), and Qwen 3-4B[36] (7.28) has significantly more errors and is above the mean line (red dashed mean line). The most poorly performing are Ministral-3-3B-Instruct[9] (19.48) with significantly more transcription errors than any other model.

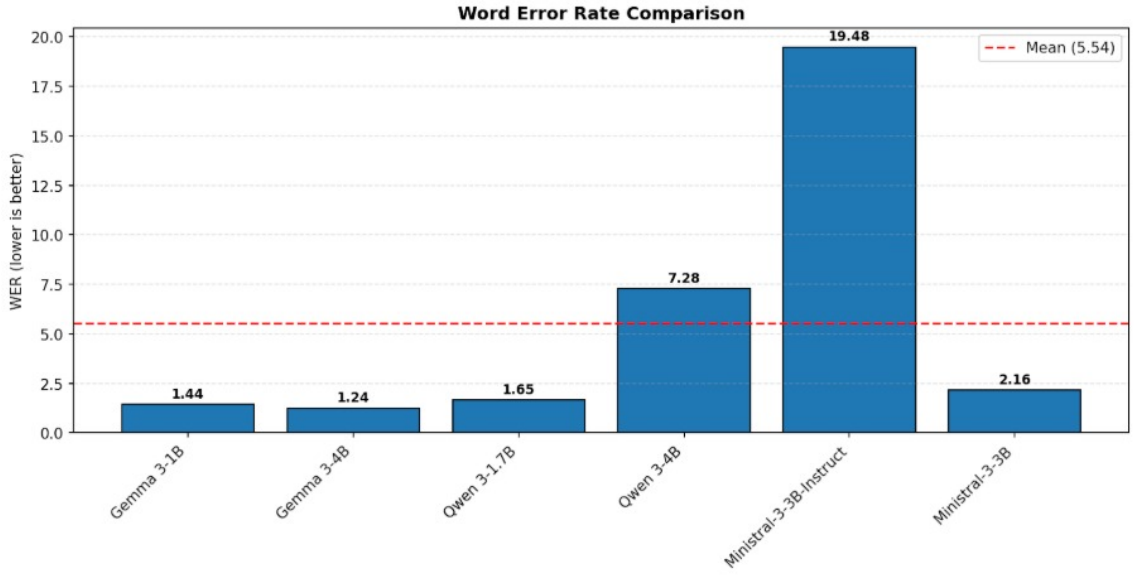

Figure 5.4: Word Error Rate (WER) Comparison Across Language Models

BERTScore measures how semantically similar a model's output is to the reference text using contextual embeddings (so higher is better), where precision reflects how much of the output matches the reference, recall reflects how much of the reference is covered, and F1 is the balanced average of both. As can be seen in the chart in Figure 5.5, Gemma 3-1B and Gemma 3-4B[36] are the most similar in an overall sense, with F1 scores the highest (approximately 0.70). Qwen 3-1.7B[44] is solid with the moderately high F1 (approximately 0.66) whereas Qwen 3-4B[44] and Ministral-3-3B-Instruct[9] has the lower overall (around 0.54-0.58) F1. The performance of Ministral-3-3B[9] is closer to the Qwen-level performance as it has better semantic similarity than the Instruct variant, better precision, recall, and F1.

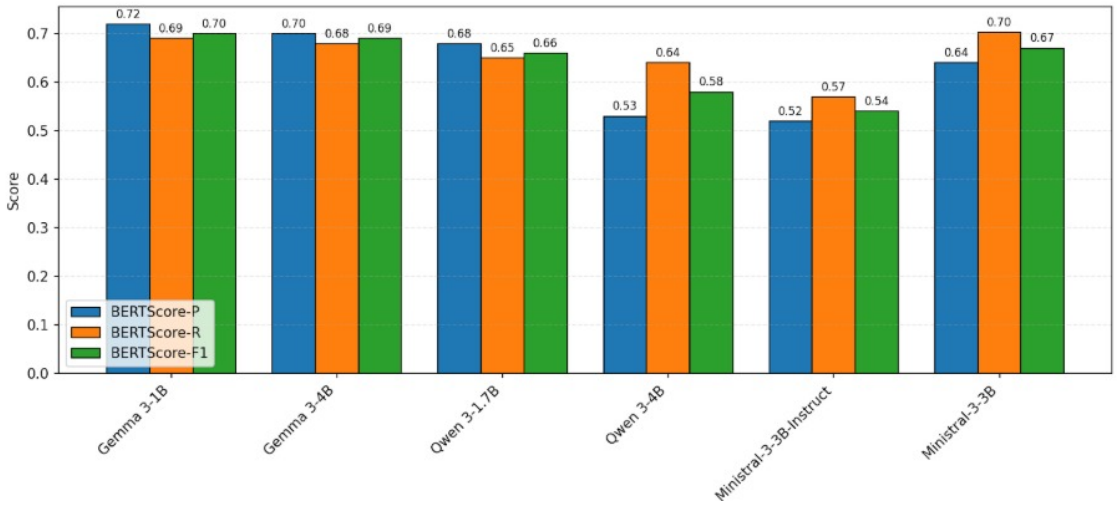

Figure 5.5: BERTScore Components Comparison Across Language Models (Precision, Recall, F1)

Cosine similarity calculates the similarity between texts as they mean (the higher the value, the more similar they are). Figure 5.6 illustrates that Gemma 3-4B[36] has the highest similarity (0.570) and Gemma 3-1B[43] the next highest similarity (0.539), indicating that the outputs of these are the most similar to the reference meaning. Qwen 3-1.7B[44] has the value of 0.458, and Qwen 3-4B[44] has 0.196. In case of the Ministral variants, the lowest similarity value is Ministral-3-3B-Instruct[9] (0.135), and Ministral-3-3B[9] takes a significantly higher similarity in the form of the cosine (0.429), meaning that it aligns the semantics well in comparison with the Instruct version.

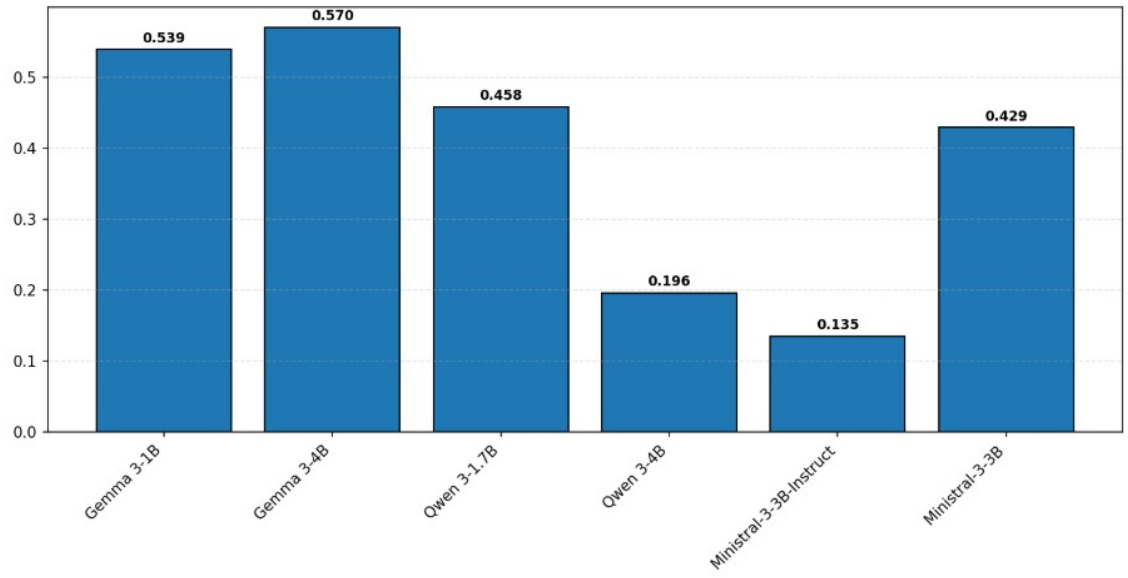

Figure 5.6: Cosine Similarity Comparison Across Language Models

In all the measures of evaluation (BLEU, METEOR, ROUGE, WER, BERTScore, and cosine similarity), the results have consistently demonstrated that the best overall performance is delivered by the Gemma models, with Gemma 3-1B[43] leading

most of the text-generation measures (BLEU, METEOR, ROUGE) and high semantic similarity (high BERTScore and cosine similarity), while Gemma 3-4B[36] performs closely and achieves the lowest WER. Qwen 3-1.7B[44] offers average and practicable performance, and Qwen 3-4B[44] declines more drastically across various metrics, especially WER and semantic similarity. The Ministral-3-3B-Instruct[9] variant is always the lowest in most measures, with Ministral-3-3B[9] scoring much better and in some instances even in the competitive mid-range. Efficiency-wise, the high efficiency of Gemma 3-1B[43] is particularly impressive given that it is able to perform at the highest or close to the highest level with reduced model size, thus it is the most feasible option given the balance between quality and cost of computation.

LLM-as-Judge Conversation Score Analysis

Gemma 3–1B: igure 5.7 shows that the overall conversation quality of the model is very high (mean = 0.860, median = 0.867), which suggests that the responses under the judge rubric are always very good. These p25/p50/p75 numbers (0.867) indicate that a majority of the conversation is concentrated at the same quality level, implying consistent behavior. Nevertheless, the range (min = 0.714, max = 0.918) suggests that there are some weaker cases, where performance falls, probably because of more difficult prompts or some cases of reasoning constraints.

Gemma 3–4B: This model is more advantageous and predictable, and this score is generally higher (mean = 0.928, median = 0.935), which makes this one closer to a great response range. The p25/p50/p75 are all equal to 0.935, which shows very similar and high-quality results over the majority of the conversations, and the p95 = 0.990 indicates that there are many perfect responses. Whereas the minimum score (min = 0.770) demonstrates that some weaker outputs still take place, the greater mean and the stronger upper-tail performance of Gemma 3-4B[36] are the clear indications that the latter offers more consistently high-quality outputs compared to Gemma 3-1B[42].

Qwen3 1.7B: The model presents good conversation quality with a mean of 0.843 and a median of 0.850, which demonstrates good to very good responses based on the judge's rubric. The identical p25/p50/p75 (0.850) values are used to indicate that the performance is concentrated and homogenous in the majority of conversations, and there is little variance in the typical cases. Nevertheless, its range of scores (min = 0.700, max = 0.900) means that there are some cases of weaker results, whereas the p95 = 0.900 demonstrates that its most successful outputs can reach the quality of the highest level.

Ministral-3 3B: This model has an average score of 0.818 and a median of 0.8245, which is comparatively poor in quality to Qwen3 1.7B[43] though it is in a good but relatively poor range. The p25/p50/ p75 values are equal (0.8245), which once again implies that there is a stable behavior in most conversations; however, the upper tail is lower (p95 = 0.873), and the minimum is also slightly poor (min = 0.679), implying more frequent failures in the quality of harder cases. In general, it seems that Ministral-3 3B[9] is still usable, but it is not as strongly manifested as it seems to be Qwen3 1.7B[43] according to the distribution of judge scoring.

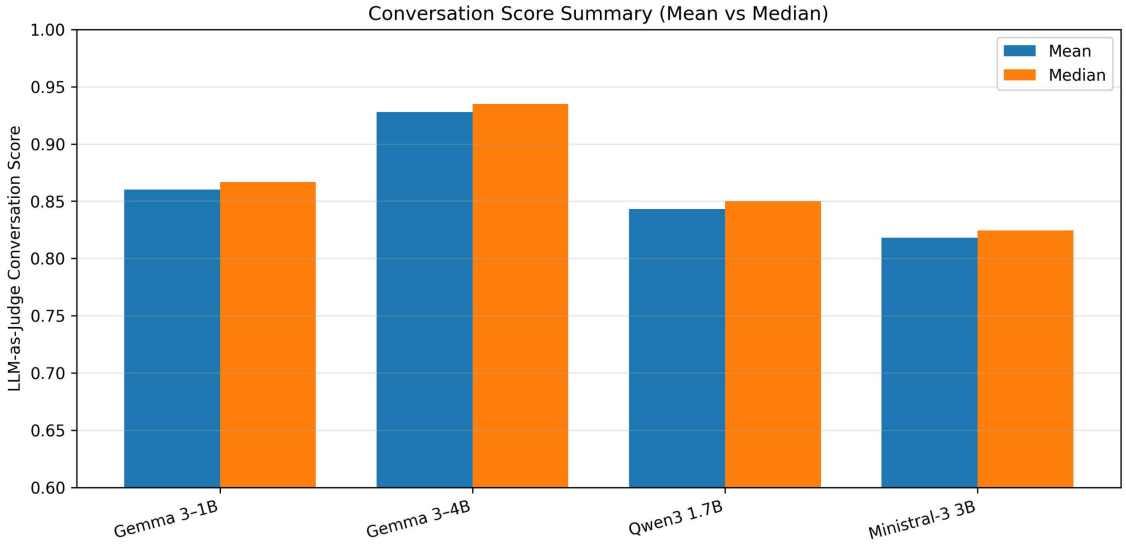

Figure 5.7: Cosine Similarity Comparison Across Language Models

Hardware Performance Analysis

In order to decide the model best suited to lightweight mobile deployment, a comparative analysis is performed under CPU-only execution through three major measures of efficiency: peak RAM usage (memory footprint), wall-time (inference latency), and average CPU utilization (compute and battery cost). A combination of these measures demonstrates the amount of memory that the model requires to operate, its responsiveness, and its load on the device processor in the generation process.

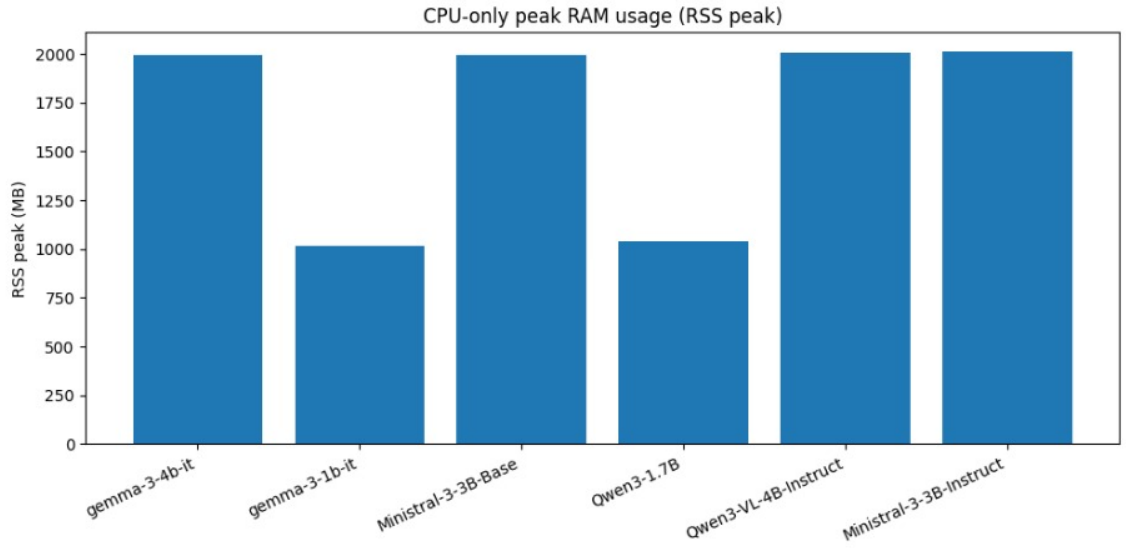

Figure 5.8: CPU-Only Peak RAM Usage (RSS Peak) Across Language Models

Figure 5.7 (the RSS peak graph) indicates the maximum RAM required by each model to perform the computation of the CPU inference- the smaller the model, the more it is supported by the mobile devices. In this case, Gemma-3-1B-it[42] has the minimal peak RAM (approximately 1 GB) and, therefore, it is the most appropriate

when deploying a lightweight solution to a mobile. The 3B-4B models, on the other hand, hit their maximum close to 2 GB, meaning they need approximately twice the memory and are not as feasible for small-RAM devices.

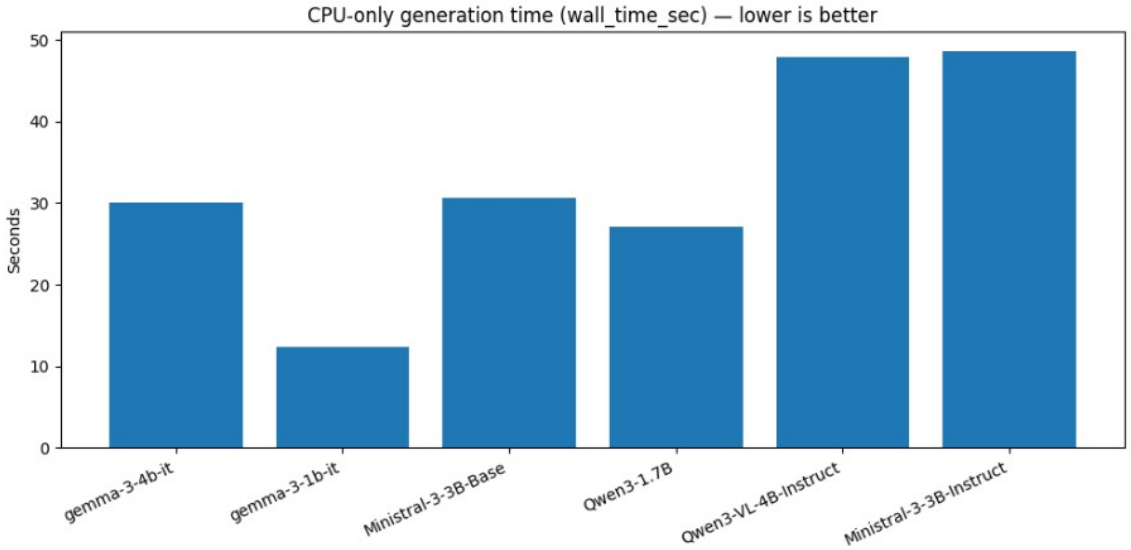

Figure 5.9: CPU-Only Generation Time (Wall Time in Seconds) Across Language Models

The wall-time plot in Figure 5.8 records the overall CPU-only time to generate text of each of the evaluated models, where lower seconds means faster inference and quicker responses, which is important for mobile usability. In the results, Gemma-3-1B-it[42] is the fastest (about 12–13s), followed by Qwen3-1.7B[43] (27s), while larger models like Gemma 3-4B-it[42] and Ministral-3-3B are around 30s, and the slowest are Qwen3- VL-4B-Instruct[43] and Ministral-3-3B-Instruct[9] near 48–49s, showing that big ger models take significantly longer to generate on CPU.

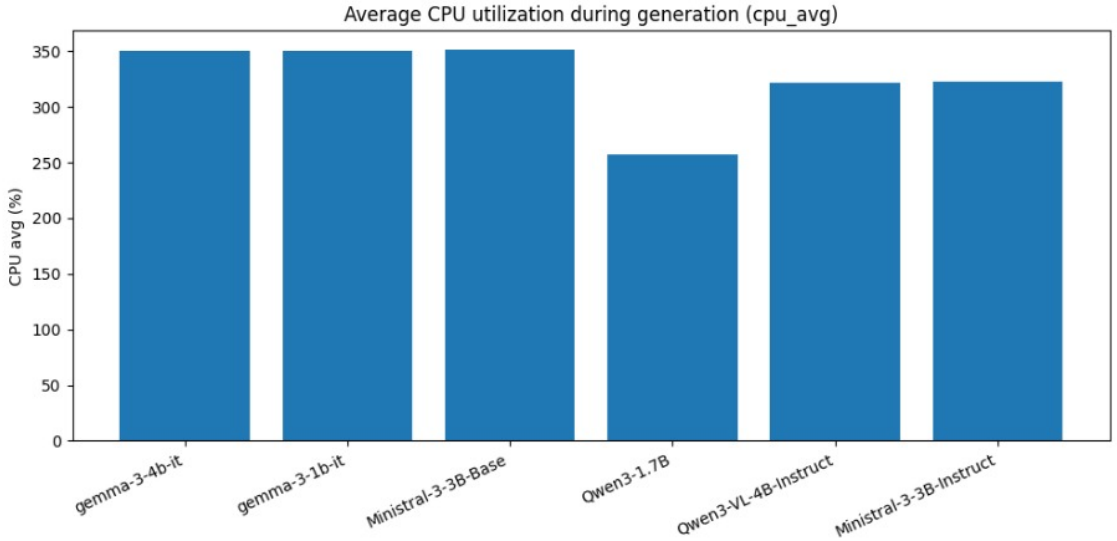

Figure 5.10: Average CPU Utilization During Generation (CPU Avg)

The CPU average utilization curve in Figure 5.9 represents the amount of CPU

power that each model uses in generation, with a lower CPU power usage being an indicator that the model is better with respect to efficiency, reduced heating, and longer battery life when deployed to a mobile environment. Results suggest that Qwen3-1.7B[43] has the lowest average CPU utilization (255%), compared to Gemma-3-1B-it[42]. Gemma-3-4B-it[42], and Ministral-3-3B-Base[9], which have higher (around 350%) ones, and Qwen3-VL-4B-Instruct[43], and Ministral-3-3B Instruct[9], which are slightly lower than that (320), meaning Qwen3-1.7B[43] is the most CPU-efficient among these during inference.

5.2.1 Hardware Performance Analysis of Image Model on CPU (Inference Latency and Stability)

These latency results clearly indicate that MobileNetV3-Small is the most suitable model on a CPU, as it remains approximately 12-18 ms with minimal variability over 60 consecutive runs, as indicated in Figure 5.10. EfficientNet-B0 is comparatively slower with an average time of about 50-80 ms, and even though it is relatively stable, it is far more time-consuming per inference than MobileNetV3- Small, which can be observed in Figure 5.10. In contrast, MobileViT-XS is the slowest and most unstable with a performance of about 100-170 ms with visible spikes (up to nearly 190 ms), not only does it have a larger delay, but its behavior is also less predictable. The same conclusion is supported by the average latency bar chart, which is shown in Figure 5.11: MobileNetV3-Small has the lowest average latency, EfficientNet-B0 is in the middle, and MobileViT-XS has the highest in the mean latency. According to the stability of run-wise latency in Figure 5.10 and the comparison of the average latency in Figure 5.11, MobileNetV3-Small is chosen as the most appropriate model for deployment in mobile hardware limitations.

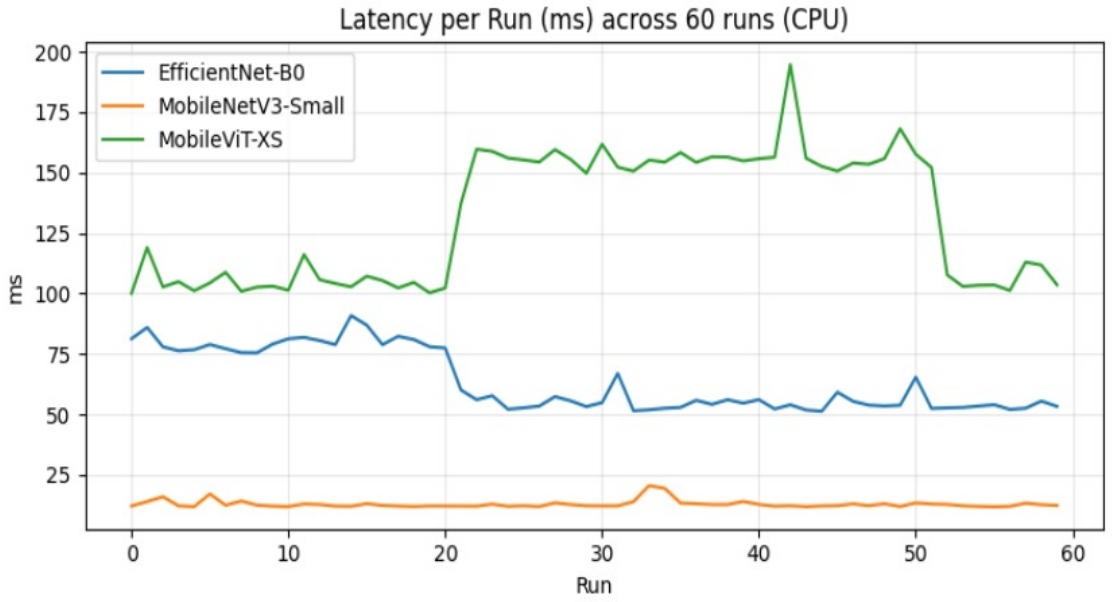

Figure 5.11: Average CPU Utilization During Generation (CPU Avg)

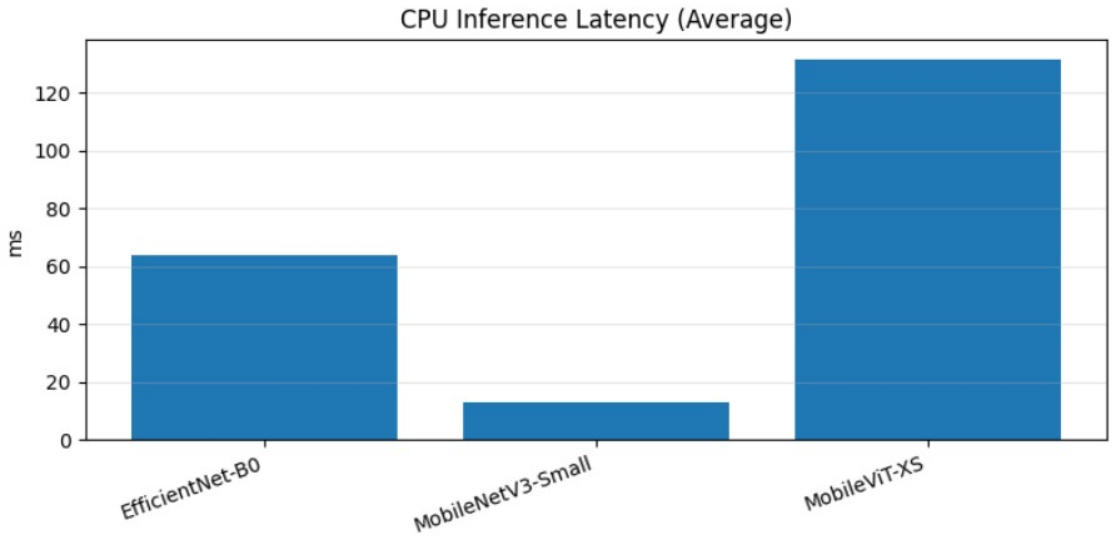

Figure 5.12: Average Mobile CPU Inference Latency Comparison (EfficientNet-B0, MobileNetV3-Small, MobileViT-XS)

The reason why MobileNetV3-Small is chosen is that it is the most feasible model to deploy in mobile, although another (MobileViT-XS) might perform slightly better. Accuracy is not the only important point on smartphones, but fast, stable, and low-power inference is also required, and MobileNetV3 is specifically created to do so. In contrast to transformer-based models which depend on heavy attention operations and can run slower on mobile CPUs/NPUs, MobileNetV3 has efficient convolution layers, which are highly optimized in mobile frameworks, and take more stable latency and reduced memory with fewer operations. On top of that, MobileViT-XS is optimized with a bigger input size (256×256) and thus imposes higher computation and energy expenditure, whereas MobileNetV3 performs optimally with 224×224 without overhead. On the whole, MobileNetV3-Small is the most suitable choice regarding real-world mobile applications: it is lightweight and reliable, and despite not being as accurate as possible, it is still suitable to perform the task.

5.2.2 Experimental STT models Setup for CPU Runtime Benchmarking

A controlled CPU benchmark was also carried out to indicate deployment under offline constraints. The evaluation conditions were maintained at the same level among models:

- CPU inference was used (no GPU acceleration)
- The decoding configuration and preprocessing pipeline were held constant per model type
- A single audio clip was used to measure response-time behavior
- Benchmark audio duration was fixed at approximately 13.73 seconds

The benchmark audio clip is exclusively utilized in the measurement of run time and sample-level analysis. Model fine-tuning was performed using a much larger Bangla audio dataset, and the benchmark clip represents a controlled runtime test rather than the full training scope.

Evaluation Metrics (Real-Time Factor, RTF)

Offline speech interaction requires responsive inference. Therefore, Real-Time Factor (RTF) was used as the primary deployment metric:

RTF = Inference Time (s) / Audio Duration (s)

Interpretation:

- RTF < 1.0: faster-than-real-time (interactive feasible)
- RTF 1.0: near real-time (borderline)
- RTF > 1.0: slower-than-real-time (interactive usability decreases)

Model	**Audio Duration (s)**	**Inference Time (s)**	**RTF (time/audio)**
Whisper-Tiny	13.73	6.33	0.4611
Whisper-Base	13.73	10.7887	0.7859
Whisper-Small	13.73	20.8421	1.5182
Whisper-Large	13.73	143.3139	10.438
wav2vec2-large-xlsr-53	13.73	18.0018	1.3113

Table 5.1: CPU runtime benchmark of Bangla STT models

Table 5.1 shows the CPU inference time and Real-Time Factor (RTF) of five STT models tested in the same conditions with a single 13.73-second audio file. RTF is calculated as inference time/audio duration; a value less than 1.0 means faster-than-real-time transcription, and a value greater than 1.0 means slower-than-real-time performance.

Qualitative Output Stability

A qualitative inspection of predicted transcriptions was performed to assess stability and usability in an end-user setting:

- **Whisper-Tiny:** fastest inference; however, output instability was observed in the transcript (garbled/incorrect trailing tokens), which can reduce reliability in farmer-facing voice interfaces
- **Whisper-Base:** produced stable and readable Bangla output while maintaining faster-than-real-time performance
- **Whisper-Small:** stable output, but latency exceeded real-time (RTF > 1), reducing conversational responsiveness

- **Whisper-Large:** stable output, but extremely high latency on CPU makes offline interactive usage impractical
- **wav2vec2-large-xlsr-53:** slower-than-real-time inference (RTF = 1.3113) and a very high sample WER on the benchmark clip, indicating poor alignment with the reference transcript under the tested decoding/normalization setup

5.3 Final Design Adjustment

Justification for Selecting Gemma Core LLM, Image Model, STT, TTS:

According to a multi-metric analysis, the best language model in the proposed lightweight compound system that is to be deployed on the mobile phone in a rural Bangladesh area could be Gemma 3-1B[42]. On typical text generation and semantic evaluation measures, such as BLEU, METEOR, ROUGE, BERTScore, cosine similarity, and WER Gemma 3-1B[42] consistently perform better or almost similarly to larger and competing models. Remarkably, it scores the best in BLEU, METEOR, and ROUGE, meaning that it has a high rate of lexical accuracy, content overlap, and contextual coherence, which are vital in producing credible and decipherable farming advice. The fact that it has a high BERTScore and cosine similarity also shows that the model is effective in preserving semantic meaning, and thus ensures generated responses closely match expert references instead of generating shallow and deceptive text.
Consideringreal-world deployment, Gemma 3-1B[42] provides a very good trade-off between computational efficiency and performance. Although Gemma 3- 4B[42] is highly competitive and has the lowest WER, the difference in performance is quite small in comparison to the enormous growth in the model size and resource demands. Gemma 3-1B[42], on the other hand, provides high-quality language with significantly less memory footprint, faster inference, and less energy usage, which are the main tradeoffs in low-cost smartphones, intermittent connectivity, and offline-first operation in rural farming areas. Gemma 3-1B[42] offers the strongest, most efficient, and scalable solution compared to Qwen and Ministral variants that demonstrate weaker semantic alignment, have higher error situations, or unreliable consistency. Hence, in terms of technical capability as well as practicality, Gemma 3-1B[42] is chosen as the central LLM to the future multimodal agricultural assistance system that will provide correct, significant, and accessible support to Bangladesh rural farmers. MobileNetV3-Small is chosen as the preferred image backbone because it is more efficient and stable on mobile CPUs, which are essential in deployment in the real-world on low-cost smartphones of rural farmers. As it was experimentally demonstrated, MobileNetV3- Small has the lowest and the most consistent inference latency, allowing one to make real-time predictions rapidly and reliably with little variation across runs. Models like MobileViT-XS have slightly higher accuracy, but have much larger and more unstable latency and are therefore not as appropriate in resource-constrained mobile environments. MobileNetV3-Small was created to support edge and mobile computing and provides the best trade-off between computation and energy usage and reasonable accuracy, hence it is the most viable option to use on-device agricultural disease detection. Now, in the case of offline, interactive speech processing on mobile CPUs, there were a number of tested

models, some of which proved inappropriate because of the limitations of efficiency and reliability. Whisper Large is incompatible with mobile usage because it is much slower than real-time (RTF = 10.438) and has a very large storage footprint of about 3-4 GB, resulting in undesirable response time latency and making real-time interaction infeasible with typical smartphones, although it has the lowest error rates of those tested (WER = 0.176, CER = 0.046). Wav2vec2-large-xlsr-53 fails to achieve offline interaction, and is slower than real time on CPU (RTF = 1.3113) and has a large sample-level word error rate and character error rate (WER = 0.8113, CER = 0.3333), which are an indication of unstable and inaccurate. Likewise, Whisper-Small is slower than real time (RTF = 1.5182), i.e., transcription is slower than the audio input length, producing apparent latency and decreased responsiveness in voice-driven processes, although it has relatively high transcription quality (WER = 0.2353, CER = 0.0575). All these restrictions ensure that these models cannot be deployed to real-time or offline mobile CPU deployment without considerable optimization or hardware acceleration. The model chosen to be used as the most appropriate offline mobile CPU deployment is Whisper-Base since it has the best overall trade-off of speed, accuracy, and resource use. It works with high speed than real-time (RTF = 0.7859), it is responsive to voice interaction, and also produces stable and reliable transcripts in the conditions of a CPU alone (WER = 0.353, CER = 0.092). Moreover, its storage capacity (moderate 200-300 MB) is relatively low and can be installed on a typical smartphone without putting undue strain on it in terms of its memory and storage capacity. All these properties combined make Whisper-Base a good fit in the real world, offline, voice-driven applications. Whisper-Tiny may be used as a fallback to lower-end devices with stricter hardware requirements since it can achieve the highest inference speed (RTF = 0.4611) and its smaller footprint; however, its comparatively unstable transcription quality (WER = 0.941, CER = 0.4828) would be best used in situations where responsiveness is more important than transcription quality. To support voice output, the mobile application used Flutter to interface with the Android default Text-to-Speech (TTS) engine, meaning that the mobile application did not need any extra cloud or external dependencies to generate natural Bangali audio feedback.

5.3.1 Contributions of the Proposed Compound Multimodal System

The proposed system is new because it represents a multimodal system built into a mobile phone by developing the individual components to be mobile-native by using a clear and goal-driven deployment policy. In the case of image-based diagnosis, the vision model is run as a PyTorch Mobile Lite Interpreter (`.ptl`) artifact by turning MobileNetV3 into TorchScript and using `optimize_for_mobile`. This eliminates Python dependency, freezes inference behavior to perform consistent execution, and uses mobile-oriented graph and runtime optimization to achieve fast and fully on-device visual inference. To generate language, the LLM is compiled as a single GGUF file (e.g., `Q2_K`) to allow it to run efficiently in an on-device runtime written in `llama.cpp`-style; this format supports quantizing weights (aggressively) and compressed distribution of weights, tokenizer, and model metadata, making its size and memory requirements suitable to run on a smartphone. To implement speech recognition, the STT component is implemented in a GGML-family format such

that the quantized operation of tensors and stream decoding is executed effectively on mobile CPUs at a predictable cost in terms of memory consumption. In speech synthesis, the system uses the Android text-to-speech engine by default instead of adding an extra neural vocoder onto the device, maintaining natural feedback, but reducing power usage, and making it work with a wide range of devices. The combination of these design decisions enables offline interaction with multimodal and low-latency privacy-preserving multimodal interaction to be entirely on the phone.

5.3.2 Statistical Analysis

Overall Performance Heatmap:

The performance heatmap in Figure 5.12 offers a side-by-side comparison of all the evaluated language models in a set of measures, and each cell shows the score of a given model on a particular evaluation measure (where higher is better on BLEU, METEOR, ROUGE, BERTScore, and cosine similarity, whereas lower is better on WER). Altogether, it can be seen that the heatmap shows that the Gemma models are the most consistent and that Gemma 3-1B [42] bests the overlap-based generation metrics (BLEU, METEOR, ROUGE) and high semantic similarity (high BERTScore and cosine), whereas Gemma 3-4B [42] is competitive and has the lowest WER, which would indicate that it is more robust to transcription noise. Qwen 3-1.7B[43] performs moderately and is usable, but Qwen 3-4B[43] declines drastically, and in particular, in WER and cosine similarity. Ministral-3-3B[9] is clearly better than Ministral-3-3B-Instruct[9] on most metrics, with the instruct variant scoring poorest in general, which underscores that model choice and fine-tuning style can be critical in this assessment task to gauge semantic coincidence and overlap on the surface.

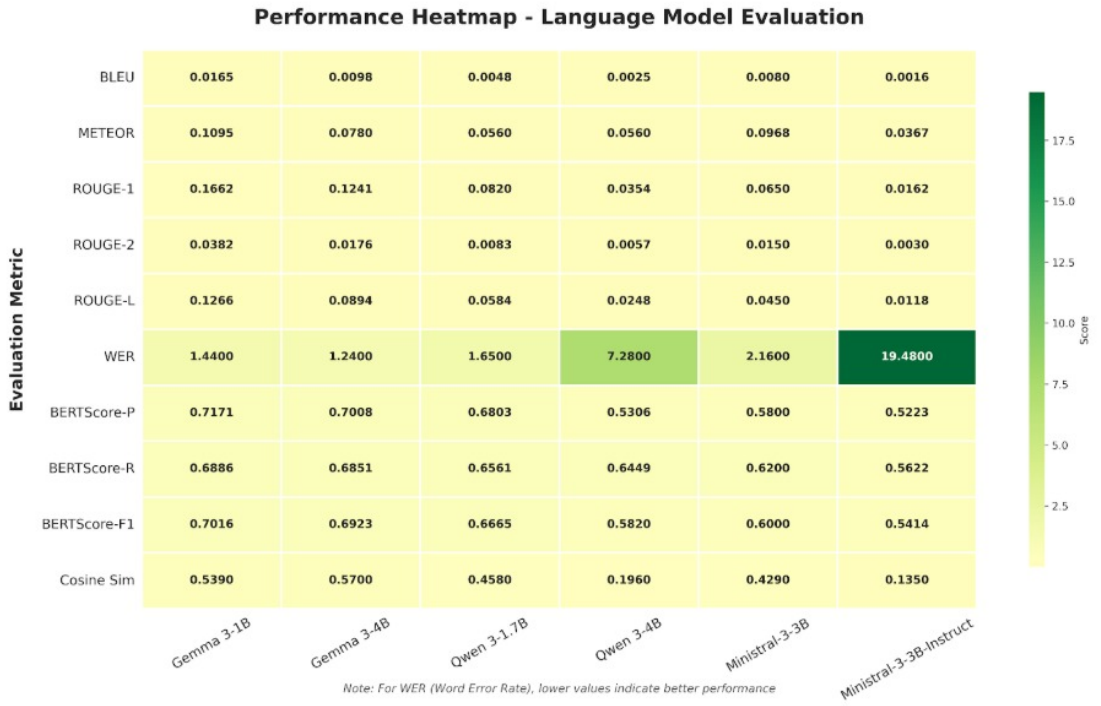

Figure 5.13: Performance Heatmap - Language Model Evaluation (BLEU, METEOR, ROUGE-1/2/L, WER, BERTScore-P/R/F1, and Cosine Similarity) across all models

Metric Correlation Matrix:

The Correlation matrix provides a summary of the nature of connections and the direction of relationship between evaluation measures, which assists in explaining the nature of agreement between various measures and the capturing of other facets of performance. Figure 5.13 indicates that BLEU, METEOR, ROUGE (1/2/L), BERTScore (P/R/F1), and cosine similarity have strong positive relationships with each other (mostly large values near +1), implying that models that perform well on one text-quality metric tend to also perform well on the other scores. Conversely, however, WER is correlated negatively with virtually all quality measurements (blue values), which shows that WER acts in the reverse direction, as an increase in WER tends to result in low scores of semantic and overlap-based quality, and WER is to be considered a metric of errors, with lower values reflecting better performance. The best correlations are seen between the variants of ROUGE (ROUGE-1, ROUGE L is nearly perfectly correlated) and between BERTScore-F1 and BERTScore-P, indicating that these families are measuring very similar behavior. Cosine similarity is also similar to BERTScore and ROUGE, indicating it serves as a good semantic agreement signal here. In general, the matrix shows that the evaluation is internally consistent, and it also shows how to motivate metrics not to be blindly brought together without managing directionality (particularly by inverting WER prior to normalization or aggregation).

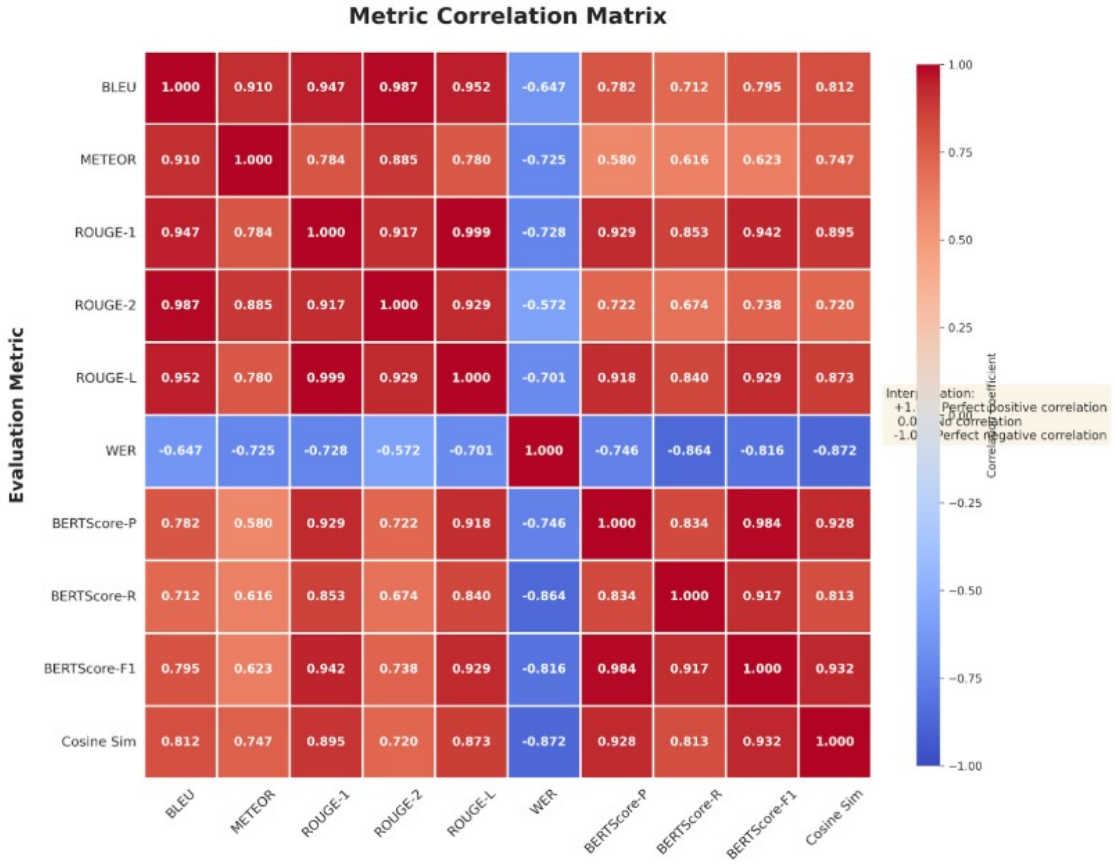

Figure 5.14: Metric Correlation Matrix of Evaluation Measures

5.4 Comparisons and Relationships

Our thesis proposes to create a Bangali-speaking, offline-first, multimodal agricultural assistant, (i) which identifies rice leaf disease on the device and (ii) which provides text and voice-based guidance to questions by the user. The system is deployed as a pipeline similar to modules: a lightweight CNN predicts the label and confidence of the disease when an image is given, and the conversation module generates disease-aware responses; when no image is given, then the assistant adheres to a question-based strategy to guess at least a likely disease context before it provides advice. This architecture is specifically addressed to rural limitations (poor connectivity, low-end devices, low literacy) and thus combines Flutter UI, local C++ STT/LLM inference services, and Bangla TTS, with all the data being completely locally available, privacy-wise, and offline reliable.

The main difference between our work and most crop disease detection research lies in the transition of a high-accuracy image classification to an implementation of a diagnostic interaction loop. The recent studies focus on accuracy optimization and reduced size of controlled datasets, for example, hybrid CNN optimization in cucumber (97.28In contrast to agricultural VQA research, our system makes a deliberate architectural compromise: instead of end-to-end multimodal fusion, we use a decoupled design where the CNN emits a symbolic disease label that conditions the language module. VQA agricultural works frequently feature multimodal co-attention, feature fusion, or LVLM fine-tuning, e.g., federated wheat-rust VQA

[22], coattention-based ILCD [29], modular co-attention with Tucker decomposition [10], and general VQA explanation systems [1]. Multimodal datasets, such as CDDM, are large-scale and contribute to pushing LVLMs with improved visual encoders [20]. We clearly verified an early feature-level fusion direction in our thesis, which created dimensional alignment problems, more expensive computation, and less flexibility, and thus is too costly to run with low resources; our workflow does not entail a complex multimodal embeddedness alignment, but can still perform image-grounded dialogue. This brings our work more in line with practical VQA behavior under constraints than it would be to maximize VQA benchmark scores, and it fills a common gap observed throughout the literature: interactive diagnosis has to be usable in cases where missing, low-quality, or inconsistent images are provided. Language constraints and modality constraints are also influential factors that our contribution heavily depends on and are not completely addressed by many chatbot/QA systems. Large agricultural chatbots and QA engines, such as high-accuracy query systems trained on millions of logs [6], [12], transformer QA enhancements [8], and LLM/RAG services deployed via WhatsApp [26], typically rely on connectivity and are not designed to operate as fully offline assistants on low-end devices. Even when LLM-centric models can achieve high zero/few-shot performance [14] or can diagnose in a multimodal chat format [25], real-world rural challenges can still be encountered: variations in the Bangali dialect, noise in the speech, and voice-first interaction. We thus propose a thesis that compares lightweight LLMs on metrics more overtly sensitive to transcription noise (WER) on top of semantic similarity (BERTScore/cosine), and that at the STT pipeline, Gemma 31B[42] is the best-balanced model with 72From a dataset perspective, our work differs from many existing VQA datasets that emphasize image availability and single-turn QA. Our dataset is designed as multi-turn diagnostic conversations in Bangla, supporting both text-only and multimodal modes, with 96,003 diagnostic entries and 9,138 rice leaf images across six classes (five diseases + healthy). It is constructed via web-scraped agricultural knowledge and curated visual data, with external expert verification for scientific correctness, and a stratified split for balanced evaluation. This directly targets the real-world scenario highlighted in our thesis discussion: farmers often begin with natural-language symptom descriptions, require clarification questions, and may not always provide images—conditions that standard image-heavy VQA benchmarks do not fully represent. Lastly, it may be helpful to compare it with AgriBuddy [44], which is also focused on the agriculture of Bangladesh but has a different systems philosophy. AgriBuddy is a RAG architecture agent-based on storing structured mappings in PostgreSQL and semantic retrieval with ChromaDB, where solutions to diseases are found in resources such as PlantwisePlus, and the vision detection is trained on the Paddy Doctor dataset (10 disease categories). It supports multilingual queries (Bangla/English/Banglish) and emphasizes retrieval + synthesis rather than purely on-device generative reasoning. In contrast, our thesis has been modeled as an entirely offline mobile system that stores disease context in the active session and applies responses to a small local LLM conditioned to the prediction of the on-device CNN prediction with as few external dependencies on databases or cloud services as possible. This creates a clear trade-off relationship: AgriBuddy can leverage broader document retrieval and richer knowledge infrastructure when storage/management layers are available, while our approach prioritizes guaranteed availability under zero connec-

tivity and minimal runtime overhead through modular activation (only triggering CNN/STT/LLM components when needed). Overall, this research is at the intersection of (i) lightweight crop disease detection, (ii) interactive VQA-like diagnostic interaction, and (iii) offline multilingual support. Rather than emphasizing only the benchmark accuracy as reported in studies of image-only classification of disease [13], [34], [40] or per the framework of the leaderboard-based VQA evaluation metrics [20], [22], [29], the emphasis is placed on a deployable, human-centered system design that follows the real-world rural ICT constraints. Such limitations are offline-first mode, Bangali voice-based communication, persistence of conversational sessions, and predetermined response behavior, which is confirmed in the end-to-end mobile deployment pipeline.

5.5 Discussion

The experimental results demonstrate that lightweight language models can effectively support multimodal, offline-capable agricultural VQA systems when combined with a task-specific CNN-based visual encoder. This evaluation demonstrates that selecting an on-device language reasoning module for rural, offline agricultural assistance requires balancing response quality, semantic faithfulness, and robustness to speech noise, rather than optimizing a single metric. All three lexical-overlap scores (BLEU, METEOR, ROUGE) demonstrate that the Gemma family (particularly Gemma 3-1B[42]) aligns most closely with reference answers, meaning that it can generate both contextually competent and structurally similar responses to expert advice. The same tendency is supported with semantic metrics (BERTScore and cosine similarity in Figures 5.5-5.6): outputs by Gemma are meaning-preserving despite changing wording, which is significant in the case of agricultural advice since there can be a variety of correct phrasings. Meanwhile, the WER comparison (Figure 5.4) reveals a voice-interface requirement: models should be robust when subjected to noise in STT transcription, and in this case Gemma 3-4B[42] has the lowest WER and Gemma 3-1B[42] is not far behind, indicating that models generally remain robust to voice-first interaction. In addition to quality, hardware constraints have a high impact on the deployment on low costs smartphone. The CPU-only based benchmarks (Figures 5.7-5.9) indicate that the Gemma 3-1B[42] possesses the best mobile profile, and its peak RAM usage is lowest, and the quickest wall-time of all the tested models, which directly aids offline, low-latency interaction on limited-memory devices. Qwen3-1.7B[43] has better average CPU utilization (Figure 5.9), though with lower conquering quality and semantic alignment in total, generating the core measures, indicating that efficiency is not sufficient to support reliable decision making. The overall trend in Figures 5.1-5.9 suggests that the scaling model size does not necessarily become more realistic in the real world: larger and instruct variants (e.g., Ministral-3-3B-Instruct[9]) not only worsen the overlap-based and semantic scores, but also become more resource-intensive and thus less feasible in offline development. The confidence of the findings is enhanced by the metric correlation matrix, which demonstrates that the majority of quality metrics fluctuate in the same direction (BLEU, METEOR, ROUGE, BERTScore, cosine), whereas WER always moves in the opposite direction, which proves that the evaluation measures both the quality of the generated text and speech robustness as two similar yet different aspects of performance. Altogether, the discussion justifies the final choice

of the design: Gemma 3-1B[43] offers the optimal quality-to-efficiency ratio of a lightweight, offline Bangali agricultural assistant, whereas MobileNetV3-Small and Whisper-Base serve as the complements of the pipeline, ensuring its responsiveness and stability properties under rural settings (limited RAM, CPU-only inference, and noisy voice inputs).

Chapter 6

Mobile Application Development and Field Evaluation

The chapter introduces the implementation of the AgroAssist mobile application and assesses its actual use in real-world conditions. It begins with a discussion of how the on-device pipeline was developed to fit an interface intended to be friendly to smartphones, and then presents results from field evaluation to assess the usability, reliability, and acceptance of the interface by farmers in the offline rural environment.

6.1 Mobile Application Development

In order to deploy the suggested agricultural VQA system in real-world applications and allow its effective implementation in practice, a completely offline-capable mobile application was built. The main idea of this application is that farmers can engage with the diagnostic system in a natural way with images and voice, without the involvement of an internet connection and expensive hardware. The mobile platform is the interface between the suggested lightweight AI and its deployment in the field-level agricultural decision support. The application was built on the cross-platform mobile development framework, Flutter, which allows creating high-performance user interfaces based on a single codebase. Flutter was chosen because it has a powerful rendering engine, performs like a native application, and has powerful multimedia processing that is necessary in capturing images, recording sounds, and providing an interactive experience in real-time. The framework is also modular in development and hence easily integrated with locally deployed inference services.

6.1.1 Application Architecture

The mobile application is designed in a modular architecture that is quite practical for the proposed system design. All functional components are independent of each other, and they share the contextual information where necessary. The software is divided into four major modules, which include user interaction, image diagnosis, speech processing, and language-based reasoning. The interaction module deals with the visual and user inputs. The clean and intuitive interface was made up of widgets by Flutter, and it is composed of chat views, image attachment indicators, session navigation panels, and control buttons. This design is easy to use for farmers who do not have much technical knowledge. The image diagnosis module will allow the

user to scan images of crop leaves through the device camera or pick images from the gallery. The imagepicker library does image acquisition. The picked image is resized and normalized, after which it is fed to a locally residing CNN-based classifier via the FlutterPyTorch Lite library. The model generates a logistic vector, on which the most likely disease category and confidence value are obtained. These findings are saved as the active diagnostic situation during the session.

6.1.2 Speech and Language Interaction

The application also has a speech-controlled interaction system to aid natural communication. Recording of audio input is done through the `record` library in a controlled waveform format that is friendly to speech recognition. The audio recording is temporarily stored with the help of the utilities of `path_provider` and `dart:io` libraries. The audio file is sent to a speech processing server that is running locally in C++, where the speech input is converted to text. The text that has been transcribed is sent to the language reasoning module. The local inference services are communicated with by use of the HTTP library via REST-style requests. The asynchronous programming model used by Flutter also ensures that these operations do not freeze the user interface and makes it responsive when it is processing. The language understanding module is also powered by a lightweight LLM-based reasoning system on a local C++ inference server. This component understands user queries, keeps conversational context, and produces disease-sensitive responses. In case there is an image-based diagnosis, the language module sets its results based on the identified disease. In case no image is given, the system will slowly collect the symptom data by interacting with the user until a likely disease situation has been determined.

6.1.3 Session Management and Persistent Memory

The application has session-based interaction and persistent memory storage to ensure that it is usable in the long run. Every discussion will be saved as an independent session with user inputs, system responses, and timestamps, as well as optional diagnostic context. The shared devices are stored in the shared location, where session information is stored as a shared-preferences library to enable the user to reuse the past discussion after the application has been restarted. Besides session history, a long-term memory feature enables users to store long-term contextual data like the type of crop, farming site, or desired cultivation methods. Such details are integrated into subsequent interactions, allowing them to respond more personally and contextually to prevent users from having to type in the same information.

6.1.4 Output and Accessibility Features

The output module displays not only the diagnostic information and guidance in text format but also in audio format. Text feedback is presented in a chat format, whereas oral feedback is created with the help of the Flutter-TTS library. This dual-mode output enhances ease of use among individuals with a low level of literacy and helps in hands-free operation in field applications. The application constantly tracks the presence of local inference services and gives visual displays to the users of system

readiness. Background timers and asynchronous status checks are used to make sure that it operates stably and to gracefully deal with temporary service interruptions.

6.1.5 Performance and Deployment Considerations

The mobile application and the supporting models were tested in a CPU-only environment in an offline condition. The modular workflow allows effective computation by having image-based disease recognition run in parallel with language-based reasoning, with very little redundant computing. The app showed solid performance, thus suitable for real-time agricultural support. The system provides data privacy and data security by working purely offline with no pictures or user requests going outside of the local device ecosystem. This deployment mode renders the application especially appropriate to rural settings, and resource-limited settings, where connectivity is inaccurate and data confidentiality is critical. Overall, this chapter outlined the design and development of a Flutter-based mobile app that realizes the lightweight agricultural VQA system proposal. Through effective image diagnosis, speech interaction, and language-based reasoning with a modular and offline capable system, the application is able to transform the research contribution into an effective tool to be used in the real-world agricultural disease diagnosis and decision support.

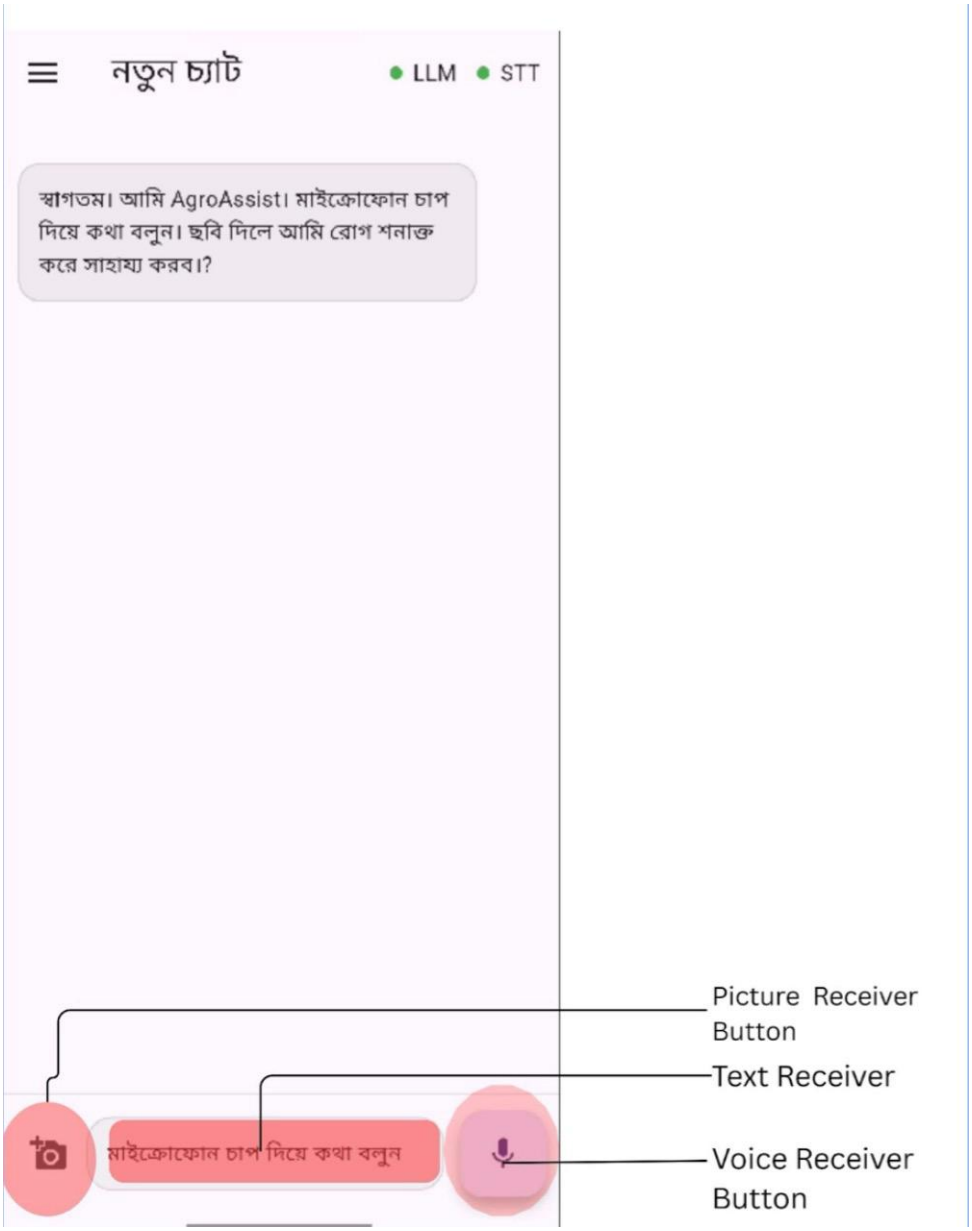

Figure 6.1: Mobile Application chat interface showing multimodal input controls (image capture button, text input field, and voice input button)

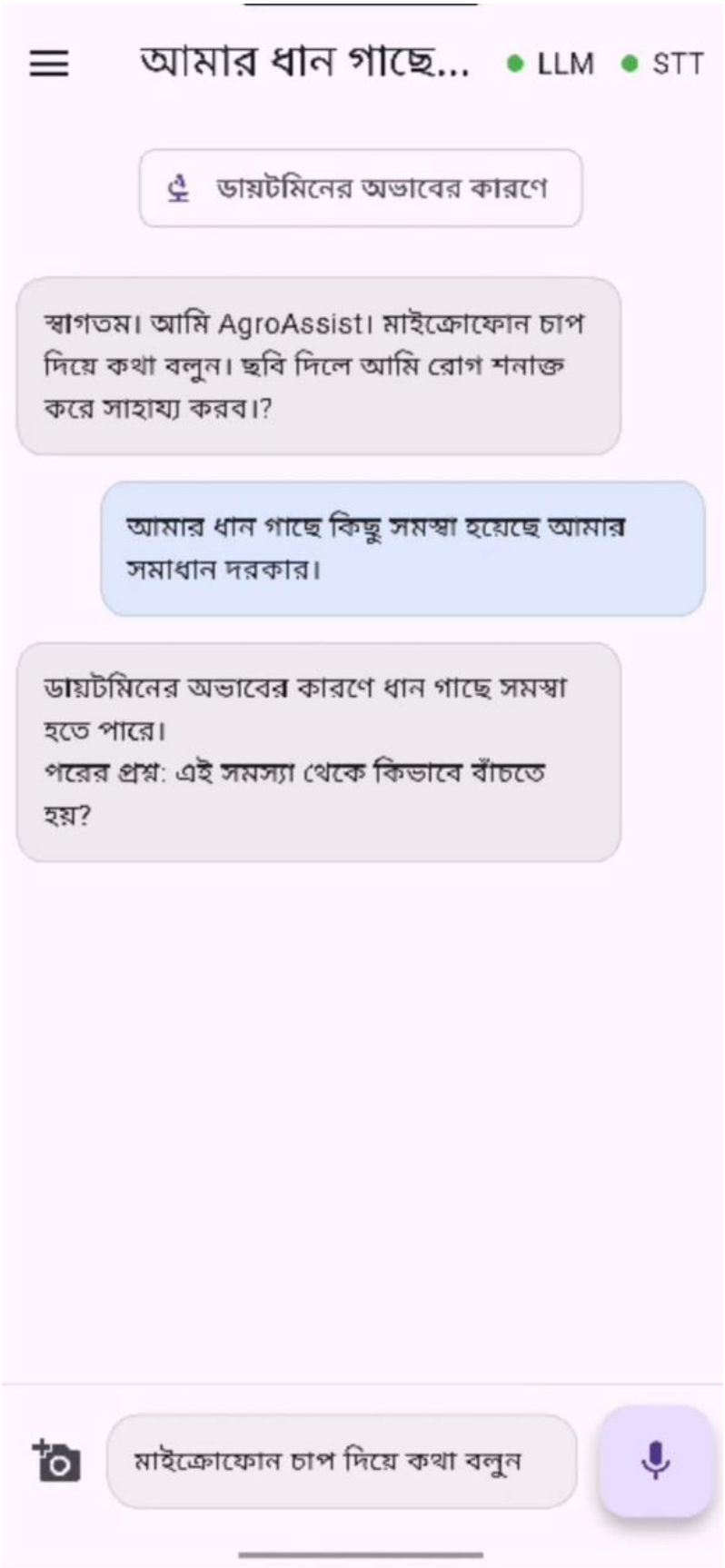

Figure 6.2: Sample Bangla conversation in AgroAssist (LLM + STT enabled)

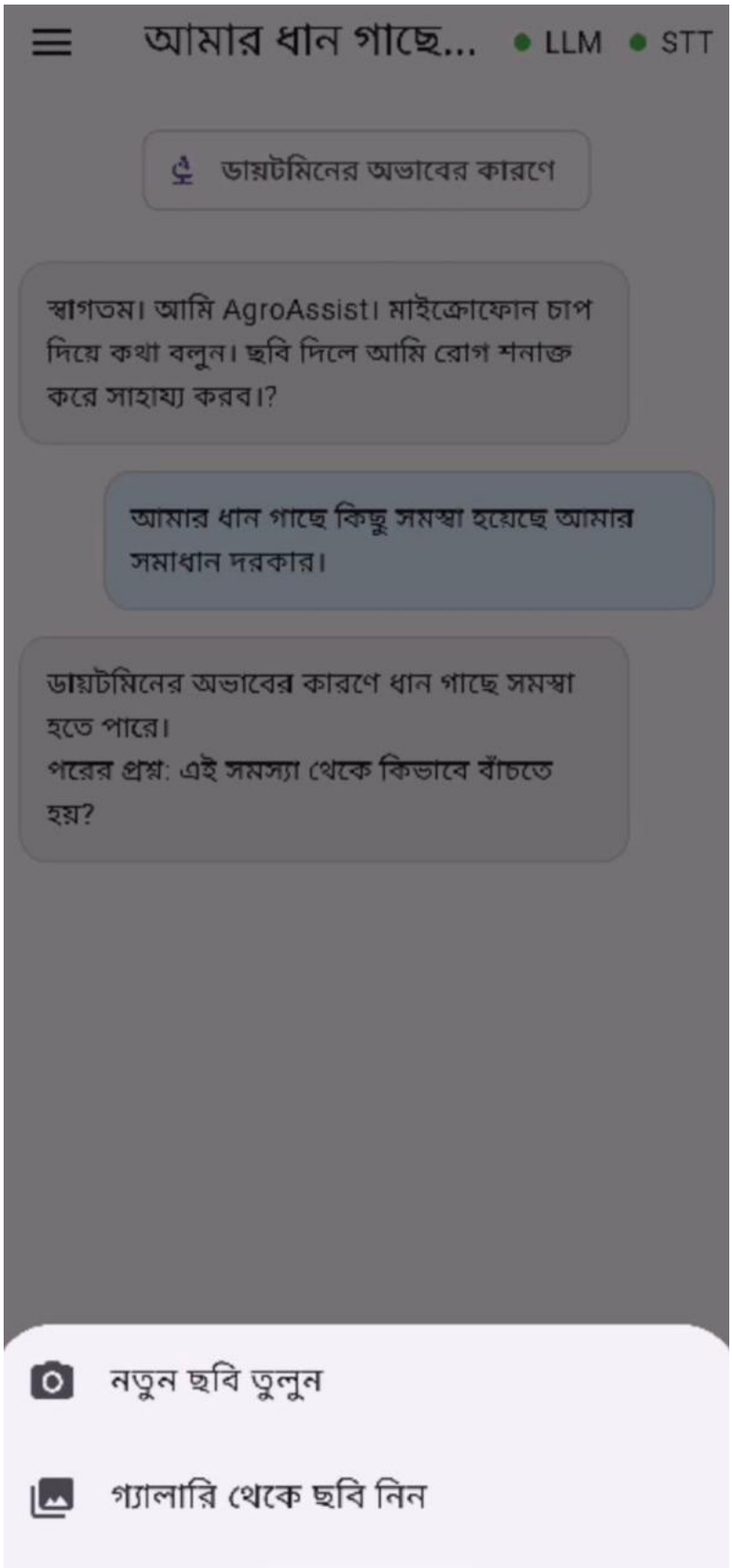

Figure 6.3: Image acquisition interface in mobile application (camera and gallery selection)

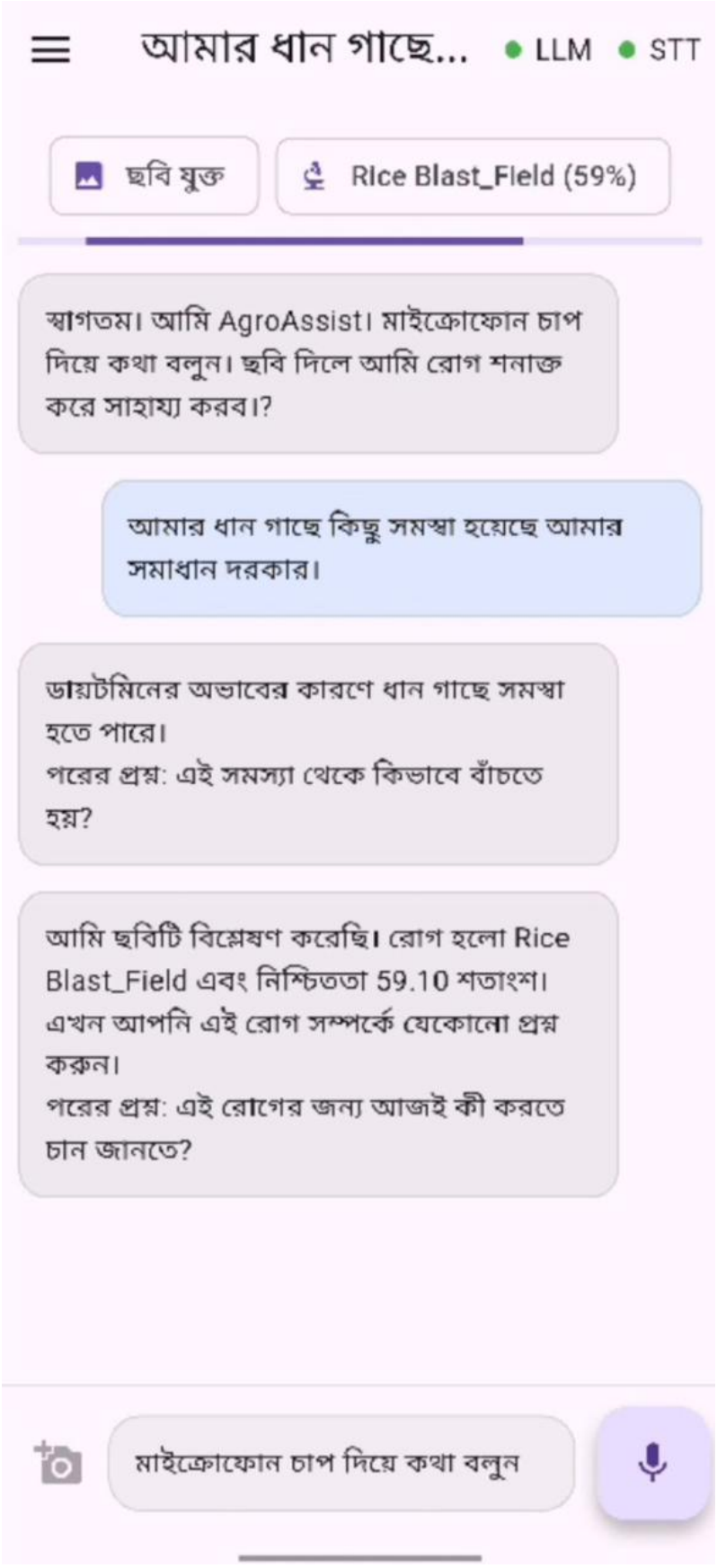

Figure 6.4: Multimodal workflow in AgroAssist (image upload → CNN prediction → Bangla response)

6.2 Field Evaluation Overview (N=30)

In order to determine the practical usability and usefulness of the proposed offline-first Bangla VQA-driven conversational agricultural assistance system (Agro Assist) in the field, a structured field test was performed with 30 farmers in several rural agricultural areas. The test was intended to be realistic to the real conditions of deploying it, with limited internet connectivity, low-cost Android phone deployment, and dialectal variation in spoken Bangla having a direct impact on system adoption and performance.
The field study was conducted in person with the various districts, where farmers were in their natural working environment (farmland and local community environment). Agro Assist was presented to the participants, who were requested to explore the application with its main functionalities, for example, Bangali conversational queries and image-based (multimodal) crop problem reporting. In addition to practical use, semi-structured interviews took place to obtain information on how the farmers would rate usability, perceived usefulness, trust, and the practical usefulness in carrying out daily farming practices. The survey questionnaire was written in Bangla and had some sections about the background of the participants, experience in using the application, language comprehension, system response quality, and qualitative feedback.

Instead of considering the assessment as a technical standard only, this paper focuses on Human-Computer Interaction (HCI) and practical deployment. It aimed at assessing whether Agro Assist can be a viable substitute for the delayed agricultural consultation and whether the system can be comfortably adopted by farmers with very little training in a rural setting.

6.2.1 Overall Acceptance and Usefulness

The results of the evaluation suggest that Agro Assist is generally accepted as a viable tool in terms of crop-related assistance. Farmers repeatedly reported that obtaining guidance through traditional channels, such as visiting the local agricultural office (স্থানীয় কৃষি অফিস), consulting input sellers, or contacting experts, often involves significant delays, ranging from several days to more than a week. Such delays may lead to an increase in the spread of the disease or pest attacks and may put the crop in danger. Conversely, the conversational accessibility of guidance available through the application was seen by the farmers as a quicker option, which allowed farmers to make a quicker decision, and reliance on a physical visit or intermediaries was diminished. One significant factor that led to this acceptance was the fact that the system was accessible in actual practice. The interaction flow in the application was perceived to be easy and user-friendly to the farmers; they could easily navigate the system with little problem, despite having less experience with their smartphones. Farmers also valued the fact that the solution complies with the rural realities, like unstable connectivity and time-localized farming decisions. In general, field feedback indicates that Agro Assist can be adequately aligned with the needs of farmers, offering instant advice, a simplified interaction model, and practical value whenever it is difficult to consult an expert on a particular matter.

6.3 Parameter-wise Evaluation of System Performance (N = 30)

In order to summarize the farmer's perceptions in a deployment-oriented fashion, a parameter-based assessment was performed based on five main system parameters, according to the hands-on use of the system and the interview feedback of farmers. All parameters were mapped to the questions in the Bangla survey questionnaire in such a way that reported scores would provide the direct user response and not assumptions. The parameters that were evaluated included User Interface, Compatibility, Dialect Understanding, Information Rightfulness, and Privacy.

User Interface (90%)

The User Interface parameter reflects how easily farmers can operate Agro Assist, including navigation clarity, interaction simplicity, and overall comfort while using the application. This parameter was mainly measured on the question: "অ্যাপটি চালু করা ও ব্যবহার করা কি সহজ ছিল?", which measures the first-time impression on usability that is received by the first time users in a rural environment. The satisfaction rate on the User Interface was given 90%, meaning that the majority of farmers found the application easy to activate and operate. This reinforces the principles of an efficient, lightweight, and farmer-friendly user interface, particularly in the case of users who have minimal previous experience with smartphone applications.

Compatibility (83.33%)

The Compatibility parameter is a measure of the consistency of the system to operate within the environment of real rural conditions, specifically with low-to-mid range Android-based devices and in less-than-ideal connectivity conditions. Two questions in the survey were used in the evaluation of this parameter: "ইন্টারনেট না থাকলে কি অ্যাপটি কাজ করেছে?" to assess offline operational feasibility, and "ইন্টারনেট ছাড়া কাজ করার কারণে অ্যাপটি কতটা উপকারী?" to capture the practical value of offline usage. Compatibility scored 83.33%, indicating that the system was reliable among most users in the normal rural situations. The outcome is also an indication that the offline-first design is practically useful but can be further optimised to enhance robustness when using devices with more constraints or severe connectivity constraints.

Dialect Understanding (10%)

Dialect Understanding implies the measurement of the Agro Assist ability to read and understand the local Bangali dialect of the farmer, and that dialect can be region specific vocabulary, accent, and phrasing. This was measured using the question "অ্যাপটি কি আপনার স্থানীয় বাংলা ভাষা বুঝতে পেরেছে?". Dialect Understanding scored 10%, making it the most significant limitation observed in the field evaluation. This low score indicates that dialect variation remains a major barrier to consistently accurate conversational interaction and highlights the need for dialect-aware language modeling and expanded rural dialect data collection.

Information Rightfulness (83.33%)

Information Rightfulness represents the perceived correctness and relevance of the system's answers, especially whether its guidance aligns with the farmer's crop condition and the submitted image context. This parameter was assessed using the multimodal correctness question "**অ্যাপটি কি ছবির সাথে মিল রেখে উত্তর দিয়েছে?**". Information Rightfulness scored 83.33, which meant that the vast majority of farmers believed that the responses of the system tended to correspond to their crop issues and were practical. The other discrepancy may be attributed to uncertainty at the field level in terms of unclear symptoms, inconsistent image quality, and sometimes communication failure due to dialectal differences. These findings indicate that the increase in dialect strength and image recognition will enhance the reliability of the response even more.

Privacy (100%)

Privacy shows the trust of the farmers that their questions, voice records, and images of their crops will not be shared among people. The aspect of privacy was not posed as a specific survey question; however, it was measured by the implementation guarantees of the system and trust of farmers, as indicated in the feedback of the interviews. Agro Assist has a high level of user isolation; for example, a farmer does not see a conversation of another farmer, and no interaction is published or shared. Furthermore, the discussions between users cannot be stored in the form of publicly accessible records that can be accessed by other users, significantly decreasing the possibilities of information leakage. The parameter-wise evaluation of privacy was rated 100% since the system design ensures cross-user visibility is prevented, and user interaction confidentiality is safeguarded, which is paramount in trust and sustained adoption in rural implementations.

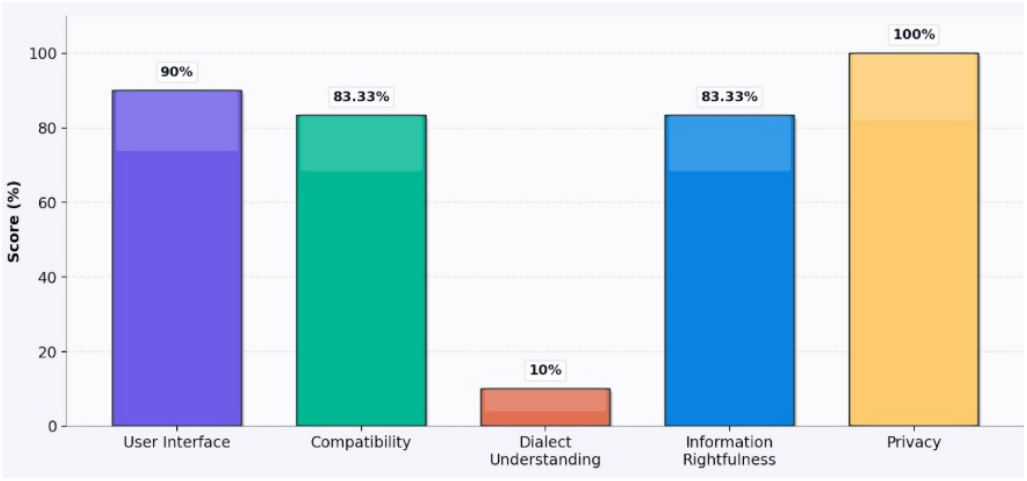

Figure 6.5: Parameter-wise Field Evaluation Scores

Combined, the parameter-wise results (Figure 6.5) demonstrate that Agro Assist scores high on User Interface (90%), Compatibility (83.33%), Information Rightfulness (83.33%), and Privacy (100%), and this means that Agro Assist is well-suited to be deployed to rural communities in real-world conditions. The first weakness is Dialect Understanding (10%), which must, however, be considered as the top

priority improvement area to be done in further versions. In general, the field testing confirms the ability of Agro Assist to deliver immediate agricultural advice and minimize the use of delayed conventional routes to consultation, and also highlights the fact that the system is secure to implement in the field as there is no retention or storage of conversations with farmers, ensuring a high level of trust, which contributes to the acceptance in the long term in rural populations.

Chapter 7

Contribution and Future Work

This chapter concludes the thesis by summing up the key results and contributions of the AgroAssist system. It summarizes the major findings of the experiment and field work that confirm the practicability of an end-to-end, offline multimodal agricultural assistant to rural Bangladesh. It subsequently describes the main research contributions regarding system design, dataset development, and evaluation methodology, and conclusively gives specific recommendations on how it could be improved in the future to become more robust, scalable, and real-world usable.

7.1 Summary of Findings

In this research, AgroAssist, a mobile-native system, was designed and validated to assist farmers by providing image-based rice leaf disease detection and Bangali conversational assistance with voice interaction, without needing internet connectivity. The final offline pipeline demonstrated that (i) MobileNetV3-Small can deliver strong on-device diagnosis performance (95.35% accuracy) with stable CPU latency (roughly 12–18 ms), (ii) Gemma 3–1B[43] provides the strongest overall response quality among the tested lightweight LLMs, and (iii) Whisper-Base enables faster-than-real-time Bangla speech recognition (RTF = 0.7859) while maintaining usable transcription quality (WER = 0.353). Besides the performance of individual models, the system-level result demonstrates that an offline, end-to-end, multimodal pipeline can be responsive enough to be useful in practice and use without violating user privacy by not relying on clouds. To be able to interact by voice, the CPU runtime benchmark (Table 5.1) indicates the reason why model choice is important concerning offline responsiveness: Whisper-Large was much slower than real-time (RTF = 10.438). Whisper-Base was faster than real time and sufficiently stable to use in conversation under CPU-only conditions. This confirms the fact that offline communication can only be done when accuracy and efficiency are balanced, particularly on the normal mobile hardware. Real-life usability was also verified through field testing in rural conditions. In the survey of the farmers (n = 30), the majority of the respondents rated the system as useful and easy to use, and most reported successful offline use and felt more confident in making decisions. The parameterwise analysis (Figure 6.5) indicated high results of the User Interface (90%), Compatibility (83.33%), Information Rightfulness (83.33%), and Privacy (100%), which suggested that the parameters were ready to be deployed practically. Nevertheless, Dialect Understanding (10%) appeared to be the most important restriction

and the most pressing point of improvement, which implies that further improvement will be evaluated in terms of core functionality, but also in terms of enhancing robustness to regional language variation and real-world speech diversity.

7.2 Contributions to the Field

This thesis makes four primary contributions:

1. **A deployable offline multimodal system for rural Bangladesh:** AgroAssist implements a full pipeline, consisting of vision-based diagnosis, Bangla speech-to-text, lightweight LLM reasoning, and Bangla text-to-speech. The system has been created with the specific intent of CPU-only and offline deployment on smartphones in privacy-preserving mode.

2. **A large Bangla conversational diagnostic dataset aligned with real farmer behavior:** The dataset includes 96,003 expert-style diagnostic entries and 9,138 rice leaf images across six classes. It is designed as multi-turn diagnostic communication as opposed to single-turn question answering, which is a more realistic model of authentic patterns of farmer consultation.

3. **A practical evaluation framework balancing response quality and deployment constraints:** Candidate models will be assessed based on lexical and semantic measures such as BLEU, METEOR, ROUGE, BERTScore, and cosine similarity. Other measures of speech robustness that have been included in the framework are the WER and CPU feasibility indicators, including inference time and real-time factor, which have a direct impact on offline usability.

4. **A mobile engineering approach for mobile-native AI components:** The system demonstrates a field of modular deployment plan with optimizing vision model programs with TorchScript or PyTorch Mobile, quantized executions in GGUF/GGML formats of on-gadget language and speech modules, and platform TTS to be approachable. This reduces application overhead while preserving an end-to-end multimodal user experience.

7.3 Recommendations for Future Work

Based on the identified gaps and capabilities of the existing AgroAssist pipeline, the following recommendations can be provided to enhance the real-world usability, robustness and scalability:

1. **Improve Bangla dialect robustness:** Since Dialect Understanding scored only 10% in Figure 6.5, future work should prioritize broader dialect coverage by expanding speech and text data across regions, introducing dialect-aware normalization, and improving STT–LLM robustness to regional vocabulary, pronunciation variation, and code-mixing. Moreover, gathering actual farmer pronunciations across various districts and assessing dialect-specific performance individually would allow specific and quantifiable progress to be made.

2. **Strengthen image robustness in real field conditions:** Remaining gaps in perceived correctness were linked to ambiguous symptoms and varied image quality. Future versions should include stronger data augmentation for lighting, blur, occlusion, and background clutter, along with clearer in-app guidance for image capture (distance, focus, angle, and leaf isolation). Adding lightweight leaf detection or quality filtering before classification can further reduce misdiagnosis caused by low-quality inputs, especially for subtle symptoms.

3. **Expand scope beyond rice leaf diseases:** Although rice is a vital crop, it would be highly beneficial to scale up the dataset and vision module to cover other crops and disease types in Bangladesh to ensure that the system is impactful and generally useful. Such growth is supposed to be done based on the same skill-based diagnostic framework to have consistent and reproducible advice on all crops.

4. **Optimize efficiency for low-end devices:** Quantization strategies, memory consumption, and end-to-end latency and energy profiling further optimization are needed to facilitate the usage of older smartphones, without compromising the quality of responses. Besides runtime feasibility (Table 5.1), battery impact testing, thermal stability analysis, and long-session performance testing should be carried out in the future.

5. **Improve overall accuracy and response reliability:** System reliability may be increased by enhancing the vision, language, and speech aspects. It is possible to increase the accuracy of vision using more real-world data, balancing the classes, and overlapping-symptom samples. Refined instruction tuning, reduction of hallucination using stricter prompting and lightweight verification, and better follow-up questioning in low-confidence conditions can help to improve the conversational reliability. STT accuracy can be increased by collecting more Bangla and regional dialect audio, improving noise robustness for field environments, and applying confidence-based re-asking or fallback-to-text mechanisms when transcription uncertainty is high.

7.4 Conclusion

This thesis introduced an offline-first, mobile-native multimodal agricultural assistant named AgroAssist in rural Bangladesh, where it was expected to operate with low connectivity, low-end smartphones, and low literacy. The system incorporates on-device rice leaf disease diagnostics, Bangali conversational advice, and voice-responsive capabilities, which allow providing end-to-end support without involving the internet and allow keeping user data entirely on-premises. One of the most important discoveries is that practical impact has more to do with the quality-efficiency trade-off than it has to do with the size of the model. Multi-metric evaluation conducted on Gemma 3-1B[43] has helped in choosing Gemma 3-1B[43] as the core LLM because of good lexical and semantic performance and mobile efficiency. MobileNetV3-Small was selected as vision due to being the most stable and fast among all the mobile CPUs for inference, and Whisper-Base was selected as STT due to its ability to carry out faster than real-time transcription and then deliver reliable results under CPU-only restriction. Android has an in-built Bangla TTS with accessible voice feedback that does not create an excessive load on the model. In addition to individual parts, AgroAssist shows the viability of an entire offline multi-modal pipeline, with a multimodal diagnostic dataset of large size and with a deployment-based assessment model, with both quality measures and runtime specifications. According to the field feedback, the system is useful and can be used offline, and interface, compatibility, usefulness, and privacy are rated highly, whereas dialect understanding is the most critical weakness and a priority area of further development.

Bibliography

[1] P. Wang, Q. Wu, C. Shen, and A. Van den Hengel, "The vqa-machine: Learning how to use existing vision algorithms to answer new questions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1173–1182.

[2] nirmalsankalana, *Rice leaf disease image dataset*, Kaggle Dataset, 2020. [Online]. Available: https://www.kaggle.com/datasets/nirmalsankalana/rice-leaf-disease-image

[3] M. F. Hossain, S. Abujar, S. R. H. Noori, and S. A. Hossain, *Dhan-shomadhan: A dataset of rice leaf disease classification for bangladeshi local rice*, version V1, Mendeley Data, 2021. DOI: 10.17632/znsxdctwtt.1 [Online]. Available: https://doi.org/10.17632/znsxdctwtt.1

[4] M. Agrawal and S. Agrawal, "Rice plant diseases detection using convolutional neural networks," *International Journal of Engineering Systems Modelling and Simulation*, vol. 14, no. 1, pp. 30–42, 2023.

[5] dedeikhsandwisaputra, *Rice leafs disease dataset*, Kaggle Dataset, 2023. [Online]. Available: https://www.kaggle.com/datasets/dedeikhsandwisaputra/rice-leafs-disease-dataset?resource=download

[6] S. Godara, J. Bedi, R. Parsad, D. Singh, R. S. Bana, and S. Marwaha, "Agriresponse: A real-time agricultural query-response generation system for assisting nationwide farmers," *IEEE Access*, vol. 12, pp. 294–311, 2023.

[7] S. I. U. Haq, M. N. Tahir, and Y. Lan, "Weed detection in wheat crops using image analysis and artificial intelligence (ai)," *Applied Sciences*, vol. 13, no. 15, p. 8840, 2023.

[8] Y. Huang, J. Liu, and C. Lv, "Chains-bert: A high-performance semi-supervised and contrastive learning-based automatic question-and-answering model for agricultural scenarios," *Applied Sciences*, vol. 13, no. 5, p. 2924, 2023.

[9] A. Q. Jiang et al., "Mistral official," *CoRR*, vol. abs/2310.06825, 2023. DOI: 10.48550/ARXIV.2310.06825 arXiv: 2310.06825. [Online]. Available: https://doi.org/10.48550/arXiv.2310.06825

[10] Y. Lan, Y. Guo, Q. Chen, S. Lin, Y. Chen, and X. Deng, "Visual question answering model for fruit tree disease decision-making based on multimodal deep learning," *Frontiers in Plant Science*, vol. 13, p. 1 064 399, 2023.

[11] G. Lu et al., *Agi for agriculture*, 2023. arXiv: 2304.06136 [cs.AI]. [Online]. Available: https://arxiv.org/abs/2304.06136

[12] A. Marla, R. Paul, A. K. Saha, N. K. Basha, and B. Anandhakrishnan, "An agrobot: Natural language processing based chatbot for farmers," in *2023 4th International Conference on Smart Electronics and Communication (ICOSEC)*, 2023, pp. 1235–1241. DOI: 10.1109/ICOSEC58147.2023.10276356

[13] P. S. Thakur, T. Sheorey, and A. Ojha, "Vgg-icnn: A lightweight cnn model for crop disease identification," *Multimedia Tools and Applications*, vol. 82, no. 1, pp. 497–520, 2023.

[14] B. Zhao, W. Jin, J. Del Ser, and G. Yang, "Chatagri: Exploring potentials of chatgpt on cross-linguistic agricultural text classification," *Neurocomputing*, vol. 557, p. 126 708, 2023.

[15] D. D. Barua et al., "Chitrojera: A regionally relevant visual question answering dataset for bangla," *arXiv preprint arXiv:2410.14991*, 2024.

[16] A. Holzinger, I. Fister, H.-P. Kaul, and S. Asseng, "Human-centered ai in smart farming: Toward agriculture 5.0," *IEEe Access*, vol. 12, pp. 62 199–62 214, 2024.

[17] K. Indira and H. Mallika, "Classification of plant leaf disease using deep learning," *Journal of The Institution of Engineers (India): Series B*, vol. 105, no. 3, pp. 609–620, 2024.

[18] B. M. Joshi and H. Bhavsar, "A nightshade crop leaf disease detection using enhance-nightshade-cnn for ground truth data," *The Visual Computer*, vol. 40, no. 8, pp. 5639–5658, 2024.

[19] V. Joynt et al., "A comparative analysis of text-to-image generative ai models in scientific contexts: A case study on nuclear power," *Scientific Reports*, vol. 14, no. 1, pp. 1–23, 2024.

[20] X. Liu et al., "A multimodal benchmark dataset and model for crop disease diagnosis," in *European Conference on Computer Vision*, Springer, 2024, pp. 157–170.

[21] Y. Lu et al., "Application of multimodal transformer model in intelligent agricultural disease detection and question-answering systems," *Plants*, vol. 13, no. 7, p. 972, 2024.

[22] A. Nanavaty et al., "Integrating deep learning for visual question answering in agricultural disease diagnostics: Case study of wheat rust," *Scientific Reports*, vol. 14, no. 1, p. 28 203, 2024.

[23] R. P. Porfírio, R. N. Madeira, and P. A. Santos, "Agriuxe: Integrating explainable ai and multimodal data for smart agriculture," in *2024 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI)*, vol. 1, 2024, pp. 1–6. DOI: 10.1109/ISSI63632.2024.10720487

[24] M. F. Shahid, T. J. Khanzada, M. A. Aslam, S. Hussain, S. A. Baowidan, and R. B. Ashari, "An ensemble deep learning models approach using image analysis for cotton crop classification in ai-enabled smart agriculture," *Plant methods*, vol. 20, no. 1, p. 104, 2024.

[25] K. Sharma, V. Vats, A. Singh, R. Sahani, D. Rai, and A. Sharma, "Llava-plantdiag: Integrating large-scale vision-language abilities for conversational plant pathology diagnosis," in *2024 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2024, pp. 1–7.

[26] N. Singh et al., "Farmer. chat: Scaling ai-powered agricultural services for smallholder farmers," *arXiv preprint arXiv:2409.08916*, 2024.

[27] D. Sonmez and A. Cetin, "An end-to-end deployment workflow for ai enabled agriculture applications at the edge," in *2024 6th International Conference on Computing and Informatics (ICCI)*, IEEE, 2024, pp. 506–511.

[28] L. Wang, T. Jin, J. Yang, A. Leonardis, F. Wang, and F. Zheng, "Agri-llava: Knowledge-infused large multimodal assistant on agricultural pests and diseases," *arXiv preprint arXiv:2412.02158*, 2024.

[29] Y. Zhao et al., "Informed-learning-guided visual question answering model of crop disease," *Plant Phenomics*, vol. 6, p. 0277, 2024.

[30] M. Alam, M. J. I. Tutul, M. A. H. Wadud, M. J. Hossen, and M. Mridha, "Bilingual bangla ocr for rural empowerment: Detecting handwritten queries and agricultural assistance," *IEEE Open Journal of the Computer Society*, 2025.

[31] K. Ashok, J. K. Mattew, K. Chandana, G. Maharathi, G. Abhilash, and N. Somani, "Smart agriculture: A comprehensive approach to crop disease diagnosis using offline multilingual technologies," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, IEEE, 2025, pp. 1874–1880.

[32] M. Awais, A. H. S. A. Alharthi, A. Kumar, H. Cholakkal, and R. M. Anwer, "Agrogpt: Efficient agricultural vision-language model with expert tuning," in *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2025, pp. 5687–5696.

[33] M. S. M. Bhuyan, E. Hossain, K. A. Sathi, M. A. Hossain, and M. A. A. Dewan, "Bvqa: Connecting language and vision through multimodal attention for open-ended question answering," *IEEE Access*, 2025.

[34] A. Bilal, J. A. Khan, A. Alzahrani, K. Almohammadi, M. Alamri, and X. Liu, "Fuzzy deep learning architecture for cucumber plant disease detection and classification," *Journal of Big Data*, vol. 12, no. 1, pp. 1–21, 2025.

[35] D. Faye, I. Diop, N. Mbaye, D. Dione, and M. M. Diedhiou, "Mango fruit diseases severity estimation based on image segmentation and deep learning," *Discover Applied Sciences*, vol. 7, no. 2, pp. 1–12, 2025.

[36] Google / Gemma team, *Gemma3-4b*, model (multimodal, 4B) on Hugging Face / Ollama, Supports multimodal inputs and 128K context window, 2025.

[37] G. Gupta and S. Kumar Pal, "Applications of ai in precision agriculture," *Discover Agriculture*, vol. 3, no. 1, p. 61, 2025.

[38] F. Liang, Z. Huang, W. Wang, Z. He, and Q. En, "Dynamic text prompt joint multimodal features for accurate plant disease image captioning," *The Visual Computer*, vol. 41, no. 8, pp. 5405–5419, 2025.

[39] H. Ma, W. Zhao, H. Yu, P. Yang, F. Yang, and Z. Li, "Diagnosis alfalfa salt stress based on uav multispectral image texture and vegetation index," *Plant and Soil*, pp. 1–19, 2025.

[40] P. Ngulube, "Leveraging information and communication technologies for sustainable agriculture and environmental protection among smallholder farmers in tropical africa," *Discover Environment*, vol. 3, no. 1, pp. 1–17, 2025.

[41] S. T. Y. Ramadan, M. S. Islam, T. Sakib, N. Sharmin, M. M. Rahman, and M. M. Rahman, "Image-based rice leaf disease detection using cnn and generative adversarial network," *Neural Computing and Applications*, vol. 37, no. 1, pp. 439–456, 2025.

[42] F. Saa-Dittoh, "Rural development and the ict4d plug-in principle for information and communication technologies," in *Integrating Indigenous and Scientific Knowledge for Sustainable Food Systems in Africa: The Plug-In Principle*, Springer, 2025, pp. 153–167.

[43] G. Team, "Gemma 3," 2025. [Online]. Available: https://goo.gle/Gemma3Report

[44] Q. Team, *Qwen3 technical report*, 2025. arXiv: 2505.09388 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2505.09388

[45] C. Zhou and X. Zhang, "Plant disease identification under imbalanced dataset using hybrid deep learning method," *Journal of The Institution of Engineers (India): Series A*, vol. 106, no. 1, pp. 19–29, 2025.