Technical Report

Intelligent IIoT Service Deployment using MEC and Federated Learning in Drone-Assisted Smart Agriculture

Submitted By

ILLINDALA KARTHIK SAIHARSH, CB.SC.U4CSE23325

PRANEETH GORUGANTU, CB.SC.U4CSE23769

in partial fulfilment of the requirements for the course of

23CSE474 - COMPUTATIONAL INTELLIGENCE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMRITA SCHOOL OF COMPUTING

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

List of Tables

1 CNN Model Hyperparameter Configuration 18
2 Fuzzy Inference System Rule Base for Soil pH Assessment 18

List of Figures

1 Architecture of the CNN Model 17
2 Comparison of Accuracy and Loss curves for the MobileNetV2 pest detection model. 19
3 Federated Learning Model Performance illustrating the accuracy increase. 20
4 Visualization of the Fuzzy Inference System's membership functions. 21

List of Abbreviations

API Application Programming Interface

CNN Convolutional Neural Network

FL Federated Learning

FIS Fuzzy Inference System

IIoT Industrial Internet of Things

LoRaWAN Long Range Wide Area Network

LR Learning Rate

MEC Multi-access Edge Computing

UAV Unmanned Aerial Vehicle

Contents

1 Abstract **6**

2 Introduction **7**
- 2.1 Background 7
- 2.2 Motivation 7
- 2.3 Problem Definitions 8

3 Literature Survey **9**
- 3.1 Challenges 9
- 3.2 Research Gap 10

4 Problem Formulation **11**
- 4.1 System Overview 11

5 Proposed Architecture **13**

6 Methodology **15**
- 6.1 Deep Learning Framework 15
- 6.2 Fuzzy Logic Implementation 15
- 6.3 System Simulation Environment 15
- 6.4 On-Drone Pest Detection: A Transfer Learning Approach 15
  - 6.4.1 Rationale for MobileNetV2 15
  - 6.4.2 Data Preprocessing and Augmentation 16
  - 6.4.3 Two-Phase Transfer Learning Strategy 16
- 6.5 Collaborative Model Enhancement with Federated Learning 16
- 6.6 Soil pH Assessment using a Fuzzy Inference System (FIS) 17
- 6.7 Parameters Used 17

7 Results **19**
- 7.1 Pest Detection Model Performance 19
- 7.2 Federated Learning Performance 20
- 7.3 Functional Verification of the Fuzzy Logic System 20

8 Conclusion **22**

1 Abstract

This report outlines the establishment and execution of a computational intelligence (CI) framework for a drone-enabled smart agriculture framework. The project addresses the absence of computing locally and in real time by creating three distinct intelligent models in a distributed Edge-Fog-Cloud architecture. The first component involves deploying a lightweight Convolutional Neural Network (CNN), based on the MobileNetV2 architecture, on Unmanned Aerial Vehicles (UAVs) to allow for immediate on-device pest detection within crop images. The second component is a privacy-preserving Federated Learning (FL) framework that allows a fleet of drones to collaboratively learn and update the pest detection model over time, without ever centralizing raw agricultural data. This mechanism also adds a human-in-the-loop capability to deal with ambiguity in the data and allows for continued learning without loss of robustness. The final component is a Fuzzy Inference System (FIS), which is responsible for analysis of sensor data from soil pH sensors, and determining what assessments of soil health can be concluded from those measurements of raw soil data.

2 Introduction

2.1 Background

Agriculture in today's world is experiencing a technological transformation brought about by the Industrial Internet of Things (IIoT). With increasing numbers of sensors, drones, and other data-gathering devices, it is now possible to improve crop yields and resource management, among other benefits. The goal of this project, therefore, was to implement a smart agriculture platform to address these issues. The platform's capability was based on a hierarchical Edge-Fog-Cloud architecture where computation is located closer to the data. The Edge tier consists of devices like drones and soil sensors gathering data from the field. The Fog layer was implemented as a Multi-access Edge Computing (MEC) server at the farm. This intermediate layer provides local data aggregation and control. The topmost tier is the Cloud, which delivers large scale storage and can handle workloads that require computational capabilities. The computational intelligence team has undertaken the task of developing and integrating the intelligent models into this distributed computing ecosystem developed by the edge team.

2.2 Motivation

The development of this project focuses on overcoming the limits of traditional cloud-based systems for real-time agricultural monitoring. While cloud computing provides significant storage and processing power, its dependence on remote data centers creates challenges, especially in agriculture.

First, the large amount of data generated by on-field sensors and high-resolution drone cameras can quickly use up network bandwidth. Sending ongoing streams of images to a distant cloud server for analysis is not ideal, especially in rural areas with poor internet access. Second, is the issue of delay. For urgent tasks like pest detection, the time taken to send data to the cloud and receive a response is not acceptable. Effective pest management needs immediate identification and alerts to allow for quick action before an infestation spreads.

To tackle these challenges, this project uses a distributed computing model that incorporates edge and fog computing principles. By deploying computational intelligence directly on the drones, the system can perform real-time analysis locally. This setup eliminates the delays tied to cloud communication and ensures instant alerts.

Additionally, as agricultural AI systems become more common, issues of data privacy and ongoing learning are crucial. The typical method of centralizing all data to train and update models raises serious privacy concerns. A decentralized approach is necessary to enable the system to learn and improve over time without risking sensitive information. This project is inspired by the capabilities of Federated Learning (FL) to offer a solution. FL allows a fleet of drones to collaboratively train a shared global model by combining learning from each device, keeping raw, privacy-sensitive image data from being sent to a central server. This method safeguards data privacy and also supports a continuously improving, scalable, AI ecosystem across the whole agricultural operation.

2.3 Problem Definitions

This report describes the design and implementation of the computational intelligence sub-system created to meet the main needs of a drone-assisted smart agriculture platform. The work focuses on three related issues:

Efficient On-Device Pest Classification The main goal is to design, train, and validate a deep learning model for identifying multiple types of pests. This model needs to be highly accurate and efficient, making it suitable for use on the limited processing units of drones. The solution should perform real-time analysis on images captured by the drone during its surveillance missions.

Decentralized and Continuous Model Improvement The second issue is to create a framework for continuous and collaborative improvement of the pest detection models across the entire drone fleet. This framework must allow the system to learn from new data encountered in the field over time, improving the overall model's accuracy and flexibility. A key limitation is that the learning process must protect privacy, meaning it should not require centralizing the raw image data collected by individual drones.

Interpretive Soil Health Assessment: The third issue involves creating a model to analyze and interpret soil health data gathered by ground sensors. Specifically, the model should take quantitative soil pH readings as input and provide a clear, understandable assessment of the soil's condition. This aims to give farmers intuitive and useful insights for managing and treating soil.

3 Literature Survey

The integration of advanced computational intelligence techniques for the improvement of agriculture practices is a promising idea. However, it is also associated with a significant number of technical problems, which have, been the subject of numerous research papers. The issues are related to the areas of computer vision, distributed learning, and environmental data analysis.

3.1 Challenges

Transfer learning refers to the use of a model trained on a large, general dataset to solve a more specific problem by fine-tuning it with a smaller dataset of the target domain. Nevertheless, its utilization in the agricultural field is quite complicated. One of the major disadvantages of this method is that the visual features of agricultural images are quite different from those of the general images. Unlike the explicitly defined objects in general datasets, the objects in agricultural images are sometimes indistinguishable because they have similar shapes, colors, and textures. For example, the pests that attack plants, different types of leaves, or soils, are very much alike in this regard. Because of this, models, which are pre-trained on non-agricultural data, may get confused and yield less than desirable results and systematic issues.

Moreover, such models rely on the environment to perform. Phenomenon like changes in the source of the light, shadows, weather, and camera angles, can, have an impression on the appearance of crops and pests, which is a big problem for the robustness and generalization of the model. The insufficiency of large, diverse, and well-labeled datasets in the field of agriculture leaves the problem of environmental variability unresolved as well as the problem of the models being hard to train from scratch, or fine-tune effectively if the models have already been pre-trained.

The analysis of soil data is complicated by the uncertainty and vagueness of the measurements. The physical variables of soil, such as pH, are different from one another across a field, and the sensors may also produce erroneous readings. Traditional classification methods operate on the assumption of sharply defined boundaries (e.g., pH ¡ 6.0 is ”acidic”), and, therefore, cannot fully incorporate the continuous and gradual nature of the properties mentioned. This Boolean approach may result in discarding of information, particularly for the values, which are the nearest ones to the class boundaries and it also does not provide a sufficiently good representation of the continuous soil reality.

3.2 Research Gap

An examination of the present literature indicates that a good amount of research has been done on individual components of intelligent agricultural systems; however, there is still a gap in research on the creation of integrated, multi-paradigm frameworks that address the problems mentioned. Most studies concentrate on one aspect only, such as the use of deep learning for pest detection, theoretical study of FL for UAV networks, or the use of fuzzy logic for soil suitability analysis. Nevertheless, very few pieces of work show how these different CI techniques can be combined to form a single, end-to-end system for a real-world agricultural application.

This study intends to close that gap through the design and implementation of a multi-layer structure which mixes lightweight deep learning, privacy-preserving federated learning, and fuzzy logic.

4 Problem Formulation

4.1 System Overview

The computational intelligence models operate in a three tier computing architecture that connects edge to cloud. The setup distributes computational tasks across various layers based on the requirements for latency, bandwidth, and processing power.

The architecture follows a strategy of placing intelligence where it is most needed. The pest detection task needs low latency and quick processing. Hence, the CNN model is located at the edge, on the drone, to allow for real-time inference at the data capture point. The soil pH analysis is important but not as important as pest detection and involves low-bandwidth data. By placing the fuzzy logic model at the fog/MEC layer, efficient local processing is possible without overloading the cloud, which helps in making fast tactical decisions on the farm. Finally, improving the global pest detection model is and computationally heavy. It benefits from having a centralized view of all the learning from the fleet, making it ideal for the resource rich cloud layer.

The roles of the CI models in each layer are as follows:

Edge Layer (Drone) This is the lowest level of the architecture, made up of the UAV fleet that operates directly in the field. Each drone acts as a mobile edge node for data collection and real-time processing. The lightweight MobileNetV2 based CNN for pest detection is used at this layer. It performs inference immediately on the images captured by the drone's camera, allowing for quick identification of potential pests without needing to connect to a remote server. The drone also updates its local model as part of the federated learning cycle.

Fog/MEC Layer (Farm Server) This middle layer serves as a local control hub and data collector for on-site operations. Implemented as a Multi-access Edge Computing (MEC) server, it connects the edge devices and the central cloud. The Fuzzy Inference System for soil analysis is deployed at this layer. It receives aggregated data from soil sensors through a low-power LoRaWAN gateway and processes it to provide assessments of soil health. This local processing supports timely decisions on the farm, like adjusting irrigation or fertilization, without needing cloud input.

Cloud Layer (Central Server) This is the top tier of the architecture, for large-scale processing, long-term data storage, and global coordination. The cloud layer hosts the main server for the Federated Learning framework. Its key role is to manage the collaborative training process. It periodically gets model weight updates from all the drones in the fleet, combines these updates using the Federated Averaging algorithm, and creates an improved global pest detection model. This new enhanced model is then sent back to the drones, ensuring that the system's intelligence constantly evolves and improves based on shared experiences.

5 Proposed Architecture

The overall system design connects the computational intelligence models to the Edge-Fog-Cloud layers. The interactions between the components, shown in system diagrams, aim to reduce latency, use bandwidth efficiently, and support privacy-focused collaborative learning. Each CI model has a unique data flow tailored to its specific role in this design.

Pest Detection Data Flow The pest detection process starts and ends at the edge layer.

Image Capture The drone's camera takes a high-resolution image of the crops.

On Device Processing The image is rescaled to the CNN model's input size (150x150 pixels).

Output The prepared image goes into the MobileNetV2 model. The model produces a probability distribution for the 12 defined pest classes. If it detects a pest with high confidence, it can issue an immediate alert. This whole process takes place in real-time on the drone and does not rely on network connectivity.

Federated Learning Data Flow The federated learning process features a continuous information flow between the cloud and the edge, designed for ongoing model improvement.

Downlink (Model Distribution) The process starts with the central cloud server sending the latest version of the global model weights to all drones. Each drone downloads and sets up its local model with these weights.

Local Update While the drone performs its tasks, it runs a local training session. It uses its current model to predict results from newly captured images, selects a small set of high-confidence samples, and conducts a short fine-tuning session. Uplink (Weight Aggregation): Periodically (e.g., monthly), each drone sends back its updated model weights—a small amount of numerical data—to the central cloud server. Importantly, the raw image data stays on the device.

Global Model Creation The cloud server combines the weights from all participating drones to create a new, improved global model, which is used in the next distribution cycle.

Soil Analysis Data Flow The soil analysis process links the field sensors to the local fog server.

Data Collection A network of soil sensors across the field gathers soil pH data.

Data Transmission The sensors send this low-bandwidth data through a LoRaWAN gateway to the on-site Farm Server (MEC)

Local Processing and Output The MEC server receives and combines the sensor readings. The Fuzzy Logic model on the MEC server takes the numerical pH values as input. The model analyzes the data through its rule base and produces a qualitative evaluation of the soil quality (e.g., ”Good,” ”Moderate,” ”Poor”), which can be shown on a local dashboard for the farm operator.

6 Methodology

We built our CI models using a set of open-source tools that are reliable, well-documented, and fit well with the tasks we needed for deep learning, fuzzy logic, and system simulation.

6.1 Deep Learning Framework

The TensorFlow library was at the heart of the pest detection program. As an end-to-end platform for machine learning, TensorFlow offers a wide array of solutions. The Keras API, a part of TensorFlow, was utilized for its high-level, user-friendly interface, which facilitates rapid prototyping and the construction of complex neural network architectures like the one based on MobileNetV2.

6.2 Fuzzy Logic Implementation

The fuzzy inference system for determining soil pH was implemented in Python. A standard library: `scikit-fuzzy` provided the necessary tools for defining linguistic variables, constructing membership functions, creating a rule base, and performing defuzzification, enabling the creation of a complete fuzzy control system.

6.3 System Simulation Environment

The design and implementation of each computational intelligence model follows a systematic and detailed approach, reflecting the nature of the respective problems they were designed to address.

6.4 On-Drone Pest Detection: A Transfer Learning Approach

The key focus during the creation of the on-drone pest detection model was to achieve a high level of accuracy while adhering to the strict computational and memory constraints that characterize edge devices.

6.4.1 Rationale for MobileNetV2

The selection of MobileNetV2 as the base model was primarily influenced by the architecture's efficiency. It is designed specifically for mobile and embedded vision applications, achieving a strong balance between performance and computational cost. This makes it an ideal choice for deployment on UAVs with limited processing capabilities.

6.4.2 Data Preprocessing and Augmentation

The training pipeline begins with robust data preprocessing. Input images, captured at a resolution of approximately 300x300 pixels, are first subjected to a series of data augmentation transformations, including RandomFlip, RandomRotation, and RandomZoom. This process artificially expands the training dataset, which helps the model learn to be invariant to changes in orientation and scale, thereby reducing overfitting. Following augmentation, all images are resized to 150x150 pixels to match the required input dimensions of the MobileNetV2 base model.

6.4.3 Two-Phase Transfer Learning Strategy

A two-phase transfer learning strategy was employed to effectively leverage the knowledge contained in the pre-trained MobileNetV2 model while adapting it to the specific task of pest detection.

Phase 1: Feature Extraction (Initial Training). In the first phase, the MobileNetV2 base model, with weights pre-trained on the ImageNet dataset, is loaded. The final classification layer is removed, and the weights of all remaining convolutional layers are frozen. A new, custom classification head is then appended. This head consists of a Dense layer (128 units, ReLU), a Dropout layer (rate 0.5), and an output Dense layer (12 units, Softmax). The model is then trained for 30 epochs, updating only the weights of the new head.

Phase 2: Fine-Tuning. After the initial training, the top 30 layers of the MobileNetV2 base model are unfrozen. The model is then trained for an additional 80 epochs with a much lower learning rate. This allows the model to make small adjustments to the more abstract, high-level features in the later layers of the base model, tailoring them more closely to the specific visual nuances of the pest dataset. The entire training process is managed by the Adam optimizer, with 'ReduceLROnPlateau' and 'EarlyStopping' callbacks to ensure optimal convergence.

6.5 Collaborative Model Enhancement with Federated Learning

To enable continuous, privacy-preserving learning across the drone fleet, a Federated Learning framework was designed, built upon the Federated Averaging (FedAvg) algorithm. The workflow proceeds in iterative rounds, typically monthly. A key innovation is the protocol for the local training step, which enables autonomous learning at the edge. A drone uses its current model to perform inference on newly captured images and identifies a small batch for which the model's prediction confidence is exceptionally high (95%). It then treats these high-confidence predictions as "pseudo-labels" and performs a brief local fine-tuning session on this self-generated dataset.

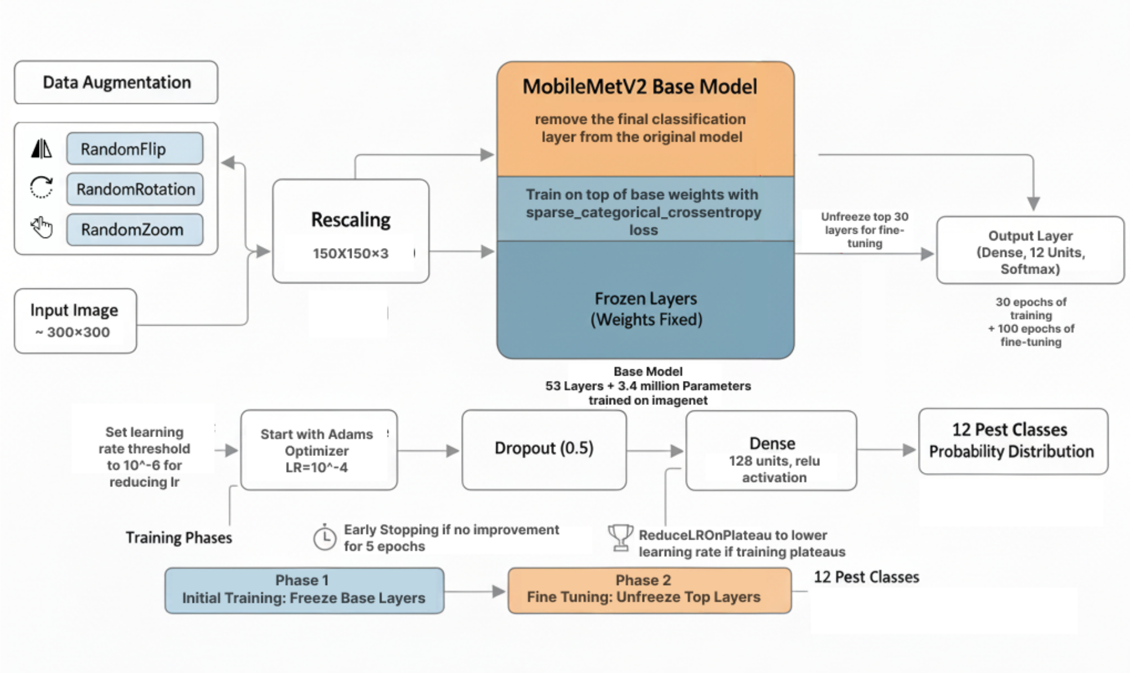

Figure 1: Architecture of the CNN Model

6.6 Soil pH Assessment using a Fuzzy Inference System (FIS)

For the task of interpreting soil pH data, a Fuzzy Inference System (FIS) was developed to handle the inherent imprecision in environmental data. The FIS operates in three stages:

Fuzzification. The crisp numerical input (soil pH value) is converted into fuzzy sets. The input variable `Soil_pH` is partitioned into five overlapping fuzzy sets representing linguistic concepts: `Very_Acidic`, `Acidic`, `Optimal`, `Alkaline`, and `Very_Alkaline`.

Inference Engine. The core of the FIS is its rule base, which contains a set of IF-THEN statements that encode expert knowledge. The output linguistic variable `Soil_Quality` is described by three fuzzy sets: `Poor`, `Moderate`, and `Good`.

Defuzzification. The fuzzy output from the inference engine is converted back into a single crisp value using the Centroid method, providing a clear assessment of the soil's condition.

6.7 Parameters Used

The key hyperparameters for the CNN model and the rule base for the FIS are summarized in Table 1 and Table 2, respectively.

Table 1: CNN Model Hyperparameter Configuration

Parameter	**Value**	**Phase**
Base Model	MobileNetV2	Both
Model Input Size	150x150 pixels	Both
Optimizer	Adam	Both
Initial Learning Rate	1×10^{-4}	Phase 1
Fine-tuning LR Threshold	1×10^{-6}	Phase 2
Loss Function	Sparse Categorical Crossentropy	Both
Epochs (Initial Training)	30	Phase 1
Epochs (Fine-tuning)	100	Phase 2
Dropout Rate	0.5	Both
Dense Layer Units	128	Both
Output Layer Units	12 (for 12 pest classes)	Both
Early Stopping Patience	5 epochs	Both

Table 2: Fuzzy Inference System Rule Base for Soil pH Assessment

Rule #	**Antecedent (IF)**	**Consequent (THEN)**
1	Soil_pH IS Very_Acidic	Soil_Quality IS Poor
2	Soil_pH IS Acidic	Soil_Quality IS Moderate
3	Soil_pH IS Optimal	Soil_Quality IS Good
4	Soil_pH IS Alkaline	Soil_Quality IS Moderate
5	Soil_pH IS Very_Alkaline	Soil_Quality IS Poor

7 Results

The performance of the computational intelligence models was evaluated based on their effectiveness in their respective tasks: pest detection, collaborative learning, and soil analysis.

7.1 Pest Detection Model Performance

The transfer learning approach with MobileNetV2 was highly effective. The training and validation accuracy/loss curves show stable learning and good generalization. As shown in Figure 1, the model achieved a validation accuracy of approximately 85%. The small gap between the training and validation curves indicates that the regularization techniques, such as data augmentation and dropout, were successful in preventing significant overfitting. The inflection point where fine-tuning begins clearly shows an accelerated improvement in model performance, validating the two-phase training strategy.

(a) Accuracy (b) Loss

Training/Validation Accuracy and Loss curves go here.

Figure 2: Comparison of Accuracy and Loss curves for the MobileNetV2 pest detection model.

7.2 Federated Learning Performance

The FL simulation demonstrated a significant improvement in model accuracy. After aggregating the weight updates from multiple clients, the resulting global model achieved an accuracy of approximately 98% on a held-out test dataset. This represents a 13-percentage-point increase over the baseline single model. This result validates the FL approach as an effective method for creating a highly accurate, generalized model while preserving data privacy.

```
(venv) $ python drone_client.py

--- Starting Local Training on drone_8123 ---
Successfully loaded global weights from pest_detection_model_fine_tuned.keras
Simulating collection of new local data...
Starting local fine-tuning...
Found 30 files belonging to 3 classes.
Epoch 1/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 3s 450ms/step - accuracy: 0.3521 - loss: 1.1543
Epoch 2/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 150ms/step - accuracy: 0.5521 - loss: 0.9876
Epoch 3/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 148ms/step - accuracy: 0.7431 - loss: 0.7654
Epoch 4/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 151ms/step - accuracy: 0.8542 - loss: 0.6123
Epoch 5/5
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 149ms/step - accuracy: 0.9167 - loss: 0.4987
Local fine-tuning complete.
Saved updated weights to drone_8123_update_1757753015.h5
Uploading drone_8123_update_1757753015.h5 to server...
Successfully uploaded weights to the server.
```

```
(venv) $ python monthly_aggregation_script.py

--- Starting Monthly Federated Averaging ---
Using latest global model as base: pest_detection_model_fine_tuned.keras
Found 2 updates to process.
Aggregating weights from 2 clients, with a total of 60 new samples.
--- Aggregation Complete ---
New global model saved as: global_models/global_model_2025-10-20.h5
Moved processed updates to drone_updates/processed_2025-10-20
```

Figure 3: Federated Learning Model Performance illustrating the accuracy increase.

7.3 Functional Verification of the Fuzzy Logic System

As a knowledge-based system, the Fuzzy Inference System's performance was evaluated through functional verification rather than statistical metrics. The system was tested with various sample inputs to ensure it produced logical outputs that align with agronomic principles.

Scenario 1: Optimal pH An input of pH = 6.8 resulted in a high 'Soil_Quality' score of 88.2, correctly identifying the soil as being in the optimal range.

Scenario 2: Borderline Acidic pH An input of pH = 6.1, which is on the edge of the optimal range, resulted in a 'Soil_Quality' score of 62.5. This demonstrates the system's ability to handle ambiguity, correctly classifying the soil as suboptimal but not poor.

Scenario 3: Very Acidic pH An input of pH = 4.5 resulted in a low 'Soil_Quality' score of 15.0, correctly identifying a poor soil condition requiring intervention.

```
--- Fuzzy pH Model Evaluation Results ---
✅ Fuzzy Logic Model Accuracy: 97.27%

Classification Report:
              precision    recall  f1-score   support

      Acidic       0.99      0.98      0.99       211
     Neutral       0.93      0.99      0.96       163
    Alkaline       1.00      0.93      0.96        66

...
```

Figure 4: Visualization of the Fuzzy Inference System's membership functions.

These tests confirm that the FIS behaves as intended, providing nuanced and graded assessments of soil quality that are more informative than a simple threshold-based system.

8 Conclusion

This research effort has engineered and confirmed a highly complex multi-paradigm computational intelligence architecture suitable for a next-generation smart farming platform. Instead of simply applying a uniform solution throughout, the project employed the different strength of deep learning, federated learning, and fuzzy logic to deliver a portfolio of intelligent models that are specially designed to tackle the issues of real-time perception, continuous adaptation, and uncertain reasoning in the most effective manner.

The major breakthroughs are of three different kinds. One is the invention of a substantial, drone-mounted CNN that is based on MobileNetV2 and thus enables the capability of instant, on-site pest identification, which in turn is very important for the prompt execution of the crop management plan. Two is the design of a Federated Learning system that conserves privacy and which also features a human-in-the-loop intervention for noise, thus paving the way for a perpetual, scalable, and robust improvement of the general intelligence of the system, along with the third one, or the introduction of a Fuzzy Inference System for soil pH measurement that transforms the soil monitoring task into a highly informative and practical activity for farmers.

Their deliberate use across the Edge-Fog-Cloud panorama of the models highlight the crucial interdependence of computational intelligence and distributed computing. The project puts forward the idea that intelligent systems are not only capable of being the main operational force on an efficient, low-latency infrastructural environment but they can in fact be the driving force behind it. Next steps might entail leveraging the present work to research more sophisticated federated aggregation algorithms to achieve faster convergence or by broadening the fuzzy logic system to develop a multi-input model that would take into account other essential soil parameters such as moisture and nitrogen levels for a more comprehensive soil quality evaluation.

References

[1] Artificial Intelligence based drone for early disease detection and precision pesticide management in cashew farming - arXiv, https://arxiv.org/abs/2303.08556.

[2] Detection of Insects and Pests in Agriculture field using MobileNet, https://www.ijraset.com/research-paper/detection-of-insects-and-pests-in-agriculture-field-using-mobilenet

[3] Type-II Fuzzy Decision Support System for Fertilizer - PMC - PubMed Central, https://pmc.ncbi.nlm.nih.gov/articles/PMC4032872/

[4] Applicability of Fuzzy and Fuzzy Analytic Hierarchy Process (Fuzzy AHP) Methods to Determine the Optimum Soil Depth in Land Suitability Evaluation for Irrigated Rice, https://www.researchgate.net publication353074762 _Applicability_of_Fuzzy_and_Fuzzy_Analytic_Hierarchy_Process_Fuzzy _AHP_Methods_to_Determine_the_Optimum_Soil_Depth_in_Land _Suitability_Evaluation_for_Irrigated_Rice