SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I5P128

Volume 12 Issue 5, 327-349, May 2025

Original Article

Edge AI-Powered System Architecture for Aloe Vera Plant Disease Detection

Sakshi koli[1*], Dev Baloni[2], Sunil Shukla[3], Anita Gehlot[1], Rajesh Singh[1], Lalit Mohan Joshi[4], Sachin Kumar[1]

[1]*Uttaranchal Institute of Technology, Uttaranchal University, Dehradun, Uttarakhand, India.*
[2]*Quantum School of Technology, Quantum University, Roorkee, India.*
[3]*School of Computing, Graphic Era Hill University, Dehradun, Uttarakhand, India.*
[4]*Haridwar University, Roorkee, Uttarakhand, India.*

[*]*Corresponding Author : Kolisakshi84@gmail.com*

Received: 19 March 2025 Revised: 20 April 2025 Accepted: 21 May 2025 Published: 27 May 2025

Abstract *- Aloe Vera has been widely cultivated for medicinal and cosmetic purposes, but its productivity is affected by different leaf diseases. They need to be detected early and accurately to avoid crop loss and to keep plants healthy. Conventional disease identification techniques are based on manual inspection, which is both time-consuming and predisposed to errors. Therefore, to cope with this delinquency, this study presents an edge AI-based system architecture for real-time aloe vera plant disease detection, which leads to more efficient and accurate detection. An Aloe Vera disease dataset was employed to train a Convolutional Neural Network (CNN), which was then deployed on an edge device to perform real-time inference. Environmental monitoring using IoT sensors is also part of the architecture. Experimental results specify that the proposed system can perceive diseases with high accuracy while considerably reducing latency compared to cloud-based methods. The proposed Aloe Vera leaf disease classification model, based on ResNet50, achieved 99.15% accuracy, 99.20% precision, 99.21% recall, and a 99.20% F1 score, ensuring high classification performance. The deployment of the quantized TFLite model on Raspberry Pi 4 B enables real-time disease detection with an inference latency of 4,922 ms (~4.9s) and a reduced model size of 23.4MB (INT8), making it suitable for edge computing applications in precision agriculture. Fine-grained deep learning with Edge AI empowers Real-Time Decision Making in Resource-Constrained Environments. This study provides a solution for Aloe Vera disease detection, characterized by low latency, energy efficiency, and scalability, emphasizing a tool for smart agriculture applications.*

Keywords *- Edge AI, Aloe vera disease detection, Deep learning, Smart agriculture, IoT-based monitoring.*

1. Introduction

Crop diseases have proven to be a stern hazard to agricultural productivity, and agriculture is very important for crops and the global economy. Diseases triggered by bacteria, fungi, viruses, and environmental stressors can wipe out plants and result in significant yield losses for smallholder farmers and even large-scale agricultural industries. Accurate and early detection of diseases is vital for reducing these losses and improving plant health and sustainable agriculture. Traditionally, plant disease detection has been performed by farmers and agricultural experts through manual inspection. Nonetheless, manual monitoring is time-consuming, requires specialist expertise, and is fragile, particularly in extensive farms. Delays in diagnosing plant diseases may lead to large outbreaks that reduce crop yield and escalate the use of chemical pesticides, which are environmentally harmful. Currently, the application of smart agriculture based on deep learning, computer vision, and the Internet of Things (IoT) is a current trend that has changed the process of plant disease recognition. Machine Learning (ML) and Deep Learning (DL) models may be utilized to annotate diseases on plant images, where very high levels of accuracy can be gained, and the fact that IoT sensors can measure environmental circumstances that lead to related diseases.

Aloe Vera is an important species in sustainable agriculture, pharmaceuticals, and cosmetics due to its potential for drought resistance, medicinal effects, and commercial value [1]. Perfect for eco-friendly farming because of its drought tolerance and low maintenance requirements. Owing to its high concentration of bioactive constituents, aloe vera has been largely used in wound healing, skin care, and digestive management [2]. Many scientific studies have shown its various biological activities, such as antiviral, antimicrobial, antifungal, and antitumour activities. Moreover, aloe vera depicts significant antidepressant activity and masterfully stands for the management of several diseases, including skin disorders (e.g., psoriasis, acne) and prediabetes [3]. Global Aloe Vera Extract Market – By Application. The global A. vera extract market has been categorized into cosmetics, pharmaceutics, and food and beverages. The market is largely dominated by the cosmetics industry (48% of the share) (Precedence Research, 2023) , owing to the moisturizing, anti-inflammatory, and skin-rejuvenating activities of aloe vera. The food and beverage industry has a 28% share and draws on the digestive and immunity-enhancing properties of aloe vera, which makes it one of the popular herbs used in health drinks and functional foods. Moreover, the pharmaceutical industry accounts for 24% of the market and uses the wound-healing, anti-microbial, and medicinal qualities of aloe vera in topical treatment and herbal medicine [4]. Increasing preference for natural and plant-based products is anticipated to bolster market growth across these industries over the coming years. The market revenue of aloe vera extract was worth over USD 2.29 billion in 2024 and is estimated to grow at a rate of more than 9.22% CAGR to be worth above USD 5.07 billion in 2033 [5]. Figure 1 illustrates the market share distribution of Aloe Vera extract by application in 2023 and highlights its usage across various industries, such as cosmetics, pharmaceuticals, and food & beverages, demonstrating its application in multiple sectors, including cosmetics, pharmaceuticals, and food & beverages.

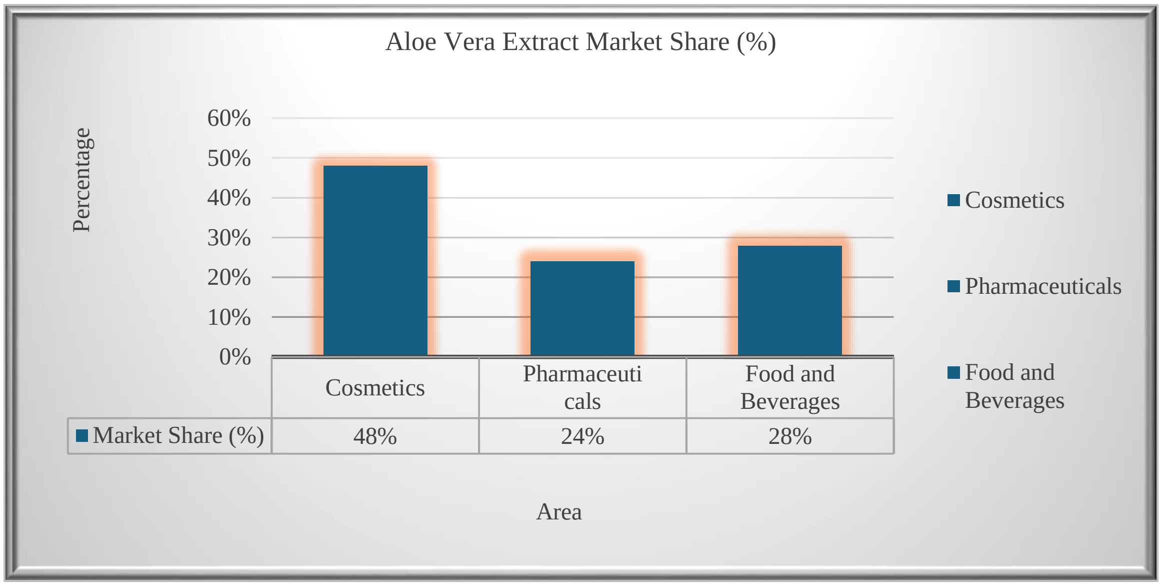


Fig. 1 Aloe vera extract market share by application (2023)

This is because of the unique morphological structure of the plant, which contains gum lodged in leaf tissue. Because of the small nature of the symptoms of the disease, it is tedious to identify infection in Aloe Vera leaves [6]. Fungal diseases are commonly diagnosed by the appearance of small spots of discoloration on leaves, which can easily resemble natural leaf aging or environmental stress. Of these, Aloe Rust is an important disease with yellow-orange spots on the leaf surface that prevent photosynthesis, weaken the plant, and make it vulnerable to nongenetic stress. If left unprocessed, this disease can cause unadorned leaf damage and hinder plant growth.

Aloe Vera plants are increasingly affected by leaf diseases that go undetected at early stages, leading to reduced yield and quality. Despite its economic importance, limited research exists on automated and accurate detection methods tailored specifically for Aloe Vera leaf diseases. Disease detection is also complicated by environmental factors such as humidity, temperature changes, and soil conditions. Current laboratory-based methods for disease identification are costly and time-consuming, making large-scale monitoring unrealistic. Although various AI-based disease detection systems have been introduced to the market, cloud-dependent models have some major limitations, such as latency, the requirement for an uninterrupted internet connection, and data security. These limitations emphasize the necessity of an automation-based real-time, rapid, and low-cost Aloe Vera crop health monitoring system. Such a system would allow the rapid diagnosis of plant diseases, which would facilitate early intervention and reduce the dependence on expensive laboratory-based approaches to diagnose plant diseases. One promising alternative is that by integrating edge computing with deep learning models, local on-device disease detection could be enabled without the need for persistent cloud connections [7]. Major Challenges in Detecting Aloe Vera Disease

- Symptom Similarity: Disease Symptoms resemble natural plant aging or response to environmental stress.
- Delayed Recognition: Pathogens may not produce noticeable symptoms in the early stages, which complicates timely intervention.
- Limitations of Manual Inspection: Manual inspection is tedious, cumbersome, and impractical for large farms.

- Disease Factors: Disease progression is influenced by humidity, temperature, and soil health
- Costly Lab Tests: Microbial testing and chemical analysis are costly and out of reach for small producers.
- Limitations of cloud AI: Delays due to Internet connectivity, data privacy issues, and increased costs.
- Insufficient Real-Time or On-Field Applications: There is a gap in developing real-time, on-field solutions for Aloe Vera disease detection, especially using edge devices for rapid diagnosis in remote agricultural settings.

Laboratory testing and AI models on the cloud are current techniques, but in most cases, they are time-consuming, costly, and not practically suitable for real-time, on-site monitoring of remote farmlands. In addition, the applications of AI for the detection of diseases in Aloe Vera are still in their infancy, with no studies focusing on real-time on-device detection and decision-making. Existing methods, such as lab tests and cloud-based AI models, are slow, costly, and impractical for real-time on-field monitoring in remote farmlands. In addition, AI-based Aloe Vera disease detection research has not yet matured, and there are no studies on real-time on-device detection and decisions. To address these challenges, this research aims to develop a novel, scalable, efficient, and intelligent Edge AI-based system for Aloe Vera leaf disease exposure. The proposed system integrates a real-time, deep learning-based disease recognition model to enable rapid and accurate identification of plant diseases. IoT-enabled environmental vision sensors are integrated to monitor critical factors influencing disease progression by capturing real-time images, and an energy-efficient Edge AI framework to facilitate climate-smart agriculture by processing data locally without relying on cloud-based infrastructures. By implementing this low-cost, high-accuracy system, this study aims to empower farmers with real-time insights, reduce crop losses, and promote sustainable Aloe Vera farming practices. The objective of this learning was to develop a strong and real-time Aloe Vera leaf disease diagnostics and prediction system based on Machine Learning & Edge AI. Divided broadly into four parts — Introduction; Literature Review; Methodology; Results & Conclusion — the research was undertaken to obtain and pre-process the images of Aloe vera leaves, training ML Models that could effectively learn to classify the diseases, and finally deploying the optimized model on an Edge AI device for its practical and real-time application. In addition, the study seeks to evaluate the system's performance against conventional approaches in terms of accuracy, effectiveness, and computational cost.

2. Literature Review

Aloe Vera, widely used in pharmaceutical and cosmetic industries, remains underexplored in disease diagnostics compared to staple crops. The disease detection of aloe vera is crucial for several reasons, including preventing crop losses, improving quality, and enabling sustainable agricultural practices. The literature review section covers various aspects related to Aloe Vera disease detection, including the design of a real-time disease recognition model based on deep learning, IoT-enabled sensors for environmental tracking, and an energy-efficient Edge AI framework for climate-smart agriculture. Since very few studies are available specifically on Aloe Vera disease detection, this review also explored research on disease detection in other plants as well. This broader review helps to identify methodologies, technologies, and approaches that can be adapted and optimized for Aloe Vera farming.

Recent advancements in IoT have enabled real-time plant health monitoring through the integration of sensors and edge devices conditions as well as possible early symptoms of disease are recorded using environmental sensors like DHT11 (for humidity and temperature), soil moisture sensors, and vision-based modules. Such systems help in making futuristic decisions in smart agriculture. In IoT-enabled agriculture, seamless communication between the edge and cloud is very important to ensure RT monitoring and decision-making. Many RF connectivity protocols are used, such as MQTT, HTTP, LoRa, Wi-Fi, ZigBee, and M2M (Machine-to-Machine) [8]. MQTT is very popular because it is lightweight messaging that has a low latency, and is also suitable for a limited resources environment. LoRa offers long-reach, low-power connectivity ideal for large distances, able to support a variety of farming needs. Wi-Fi and ZigBee provide strong medium-range local area wireless communication, enabling the integration of sensors and actuators. Additionally, M2M communication facilitates direct data transmission among devices without human intervention, which contributes to the sense of autonomy and scalability of smart agriculture. It is therefore essential not only to choose and integrate suitable communication protocols but also to guarantee that the data flow is seamless, energy-efficient, and reliable in the agricultural IoT environment [9].

For instance, the study [10] proposes an IoT-based smart arrangement for factual-time environmental nursing, automated irrigation, and plant disease prediction. It integrates microcontrollers, DHT-22 sensors, moisture sensors, rain sensors, water pumps, and VGA camera modules for continuous data collection. The OV7670 Camera Module captures plant images, which are processed using MATLAB and an AlexNet-optimised CNN for disease classification. The ML model predicts disease outbreaks based on temperature, humidity, and rainfall data, focusing on blister blight in tea plants. Implemented using an Arduino-based IoT prototype, the system enables early disease detection and precision agriculture to improve crop health.

An MMF-Net, a multi-model fusion network combining IoT and deep learning for accurate corn leaf disease detection. By integrating image and environmental data, MMF-Net achieved 99.23% accuracy, outperforming traditional models.

An IoT-based Smart Node Hub (SNH) with Arduino, sensors, and Bluetooth collects real-time soil moisture, pressure, humidity data, and temperature, transmitting it via PLX-DAQ software for continuous monitoring and instant disease alerts. This system leverages edge computing and cloud integration for efficient, scalable, and precise agricultural management [11]. An IoT-integrated machine learning model is projected for envisaging blister blight disease in tea floras using environmental data. An Arduino-based prototype with DHT-22 and rain sensors captured real-time temperature, humidity, and rainfall data. A Multiple Linear Regression (MLR) model predicts disease probability, validated through field observations from 2015 to 2019. The results showed improved prediction accuracy over time, reaching 91% in 2019. The model enables early detection, reduces pesticide use, and supports sustainable agriculture [12].

An IoT-based plant disease recognition model, where sensor nodes capture leaf imageries and transmit data to a sink node for processing. The system applies a median filter for image enhancement, followed by segmentation to identify the diseased regions. Feature extraction techniques are then used to analyze the affected areas, and a Sine Cosine Algorithm (SCA)-based Rider Neural Network (RideNN) categorizes the disease presence with improved accuracy. Simulations conducted in 50-, 100-, and 150-node IoT environments demonstrated the system's high performance, achieving 0.9156 accuracy, 0.9404 sensitivity, and 0.9298 specificity, outperforming existing models. The SCA algorithm optimizes the rider optimization algorithm (ROA) for precise classification, improving the efficiency of plant disease detection. The system supports remote monitoring, reduces labour efforts, and ensures early disease identification to minimize crop losses [13].

On the other hand, researchers propose an IoT-based framework utilizing cameras, MY THINGS smart sensors, robotic arms, and Arduino Uno for real-time data collection. Proximal soil sensors (PSS), temperature sensors, water quality sensors, and GPS assess soil fertility and environmental conditions. The system employs IoT-based communication for automated monitoring and decision-making, applying Local Binary Thresholding and Genetic Algorithm for Image Recognition to detect plant diseases. However, the system lacks disease-type classification and relies on farmer input for robotic arm control. The robotic arm automates harvesting, while GPS aids in crop spacing and irrigation [14].

To monitor potato leaf diseases, an IoT-based system is to be developed to, early blight and late blight. This system combines eco-sensing technology, image processing, and deep learning models. DHT22 for temperature and humidity measurements, and an LDR for reporting on light conditions; the system includes IoT sensors. ESP32-CAM captures images of potato leaves that are processed locally with a fine-tuned ResNet-50 model for the detection of diseases. Therefore, potato leaf diseases can be classified in real-time with an accuracy rate of 97%. The statistics from the sensors and image data are processed by the Arduino UNO microcontroller that sends SMS on the GSM/GPRS module (SIM900A) whenever the environmental parameters exceed the optimal range, for example, temperature (15-20°C) and humidity ≤ 90%. This enables farmers to receive timely alerts, thereby enhancing their ability to intervene early in disease management [15]. One study explored the feasibility of transmitting images over Low-Power Wide Area Networks (LP-WAN) using LoRa for grape leaf disease detection. Because LoRa has a low data rate and a 1% duty cycle, image transmission is challenging. The researchers optimized the image size by converting images to grayscale and tested a fine-tuned CNN model to classify grape leaf diseases. Despite packet losses of up to 50%, the model successfully identified diseases. The study demonstrates that LoRa can be used for image transmission in agriculture, with future improvements involving multiple LoRa gateways to enhance data transmission and disease detection efficiency [16].

A low-cost and energy-efficient IoT platform (SAgric-IoT) designed for real-time agricultural monitoring and leaf disease detection. It follows a five-layer architecture (gathering, communication, processing, security, and end-user) and employs ZigBee (IEEE 802.15.4) and Wi-Fi for data transmission. A Raspberry Pi 3 Model B gateway collects sensor and image data and sends them to the cloud via cellular networks. The hardware included a PIC18LF46K22 microcontroller managing AM2315, AM2302, and SHT-10 sensors for environmental monitoring, while the ESP32-CAM captured images. XBee S2C radios enable ZigBee-based communication, optimizing power and coverage. A CNN model on the cloud analyzes images with an accuracy of 95% + for disease detection. Challenges include adapting to real-field conditions owing to lighting and background variations [17]. The study proposes a Wireless IoT-based Efficient Disease Detection System (WEDDS) using WMSN with camera capability for plant disease cataloguing. The system processes images of the diseased leaves by segmenting them using a threshold-based statistical approach. Features extracted via the GLCM matrix were classified using an SVM classifier with a linear kernel. Compressed Sensing (CS) is applied to minimize data transmission overhead, and the measurements are transmitted via Raspberry Pi 3 to the cloud using ThingSpeak, leveraging inbuilt Wi-Fi. MATLAB simulations showed a cataloguing accuracy of 98.4% and a discovery accuracy of 98.5%. The system employed OpenCV for image processing and Python for implementation [18]. The paper suggests that the amalgamation of IoT, cloud computing, and big data in the agriculture domain has opened the gateway for Farm-as-a-Service (FaaS) systems for real-time monitoring, analysis, and disease prediction. FaaS takes props from IoT devices, data analytics, from predictive models to forecast the data as precisely as possible.

The IoT-Hub network featuring oneM2M and LoRa ensures stable data transmission service in low-coverage districts, and business solutions based on mobile ensure scalability. Any risk value 1 is treatable by chemical application. Combining IoT and machine learning improves prediction performance and allows automatic, real-time disease alerts for sustainable farm management [19]. The amalgamation of edge and cloud platforms delivers a vigorous resolution for enhancing system responsiveness and scalability in agricultural monitoring structures. Cloud environments, such as Google Colab, are utilized for training and optimizing deep learning models due to their high computational capabilities. In contrast, real-time inference tasks are shifted to edge devices like the Raspberry Pi 4B to minimize latency and maintain operability in offline conditions. This hybrid model addresses inherent drawbacks of cloud-dependent architectures, notably high latency, continuous internet requirements, and data security vulnerabilities, thereby assembling it idyllic for deployment in remote farming. For instance, the author [20] proposes an IoT-deep learning-based Automatic and Intelligent Data Collector and Classifier (AIDCC) framework to automate disease detection in pearl millet crops. The system integrates a Raspberry Pi, drone cameras, and environmental sensors to collect real-time data from ICAR, Mysore, India. Sensor and image data are transmitted to the cloud via Wi-Fi and MQTT, optimizing data transmission by storing up to 100 images on a Raspberry Pi before offloading. The Custom-Net deep learning model was deployed for disease cataloguing, and Grad-CAM was used for feature conception. The AI-SHES (Artificial Intelligence-based Smart Hydroponics Expert System) integrates IoT-connected sensors controlled by a Raspberry Pi to observe perilous constraints such as NPK levels, pH, turbidity, temperature, humidity, and water levels.

Sensor data are continuously uploaded to an IoT cloud server, ensuring seamless data transmission and remote access. The AI-SHES system supports both manual and automated control modes via an Android application, allowing farmers to monitor and adjust farm conditions remotely. Actuators, such as pumps, motors, and climate control systems, autonomously regulate environmental factors based on AI-driven recommendations. The system achieved 99.29% accuracy in disease detection and 99.23% F-measure in classification [21]. AI A low-slung CNN model for plant disease exposure in smart hydroponics. The arrangement is positioned on edge devices such as Raspberry Pi, integrating an energy-harvesting technique to sustain the power supply and minimize battery depletion. The Knowledge Distillation (KD) technique is applied to compress the model, reducing parameters, computations, and, consequently, power consumption. Unlike previous models that required Internet access (e.g., Google Cloud Platform), the proposed model functions offline, significantly reducing energy consumption associated with network connectivity. The proposed model achieves 99.4% accuracy while consuming only 6.22W of power, demonstrating a 2.4% accuracy improvement and a 30% power reduction compared to previous CNN models. For IoT integration, sensors and cameras capture real-time plant images, and MQTT communication optimizes data transmission to the Raspberry Pi for on-device inference [22]. This study [23] proposes crop monitoring and disease detection in a sustainable and low-cost way for smart farming by using deep learning models with IoT sensors. The system is designed to predict and detect diseases such as blast and rust in millet crops, which are essential for ensuring healthy yields. The framework employs an IoT model and device based on Raspberry Pi for disease detection and local processing, along with sensors to quantify several constraints such as temperature, humidity, and soil moisture in the field. In the results, the system performs with 98.8% accuracy, 98.2% precision, 97.4% recall, and 97.7% F-score. This framework provides a reliable and efficient model for early-stage disease detection of millet crops, which can be used to further develop sustainable farming practices, with little increase in training and testing times of 67s and 88s, respectively.

The work [24] uses smart cameras, IOT sensors, and an IOT gateway (e.g., Raspberry Pi) to detect as well as recognize plant diseases. Powered by cameras and IoT sensors, the system is used to capture environmental and plant health information, which, subsequently, is communicated using protocols such as CoAP and MQTT. The obtained data are communicated by IoT communication protocols and processed locally using edge devices like the Raspberry Pi, before transferring to a cloud-based analytics server for more in-depth scrutiny. Deep learning models using Keras and TensorFlow, including Convolutional Neural Networks, are at the heart of the system that allows for semantic segmentation of plant images. Models used include U-Net, DeepLabv3+, SegNet, and FCN-8s, with Conditional Random Fields (CRF) post-processing to smooth the segmentation and increase the accuracy of disease recognition. The SegNet model showed good performance with the MIoU of 79 % for the MIoU, and after CRF post-processing, the performance was improved. However, such a system has its constraints, and one of them is the requirement for additional dataset augmentation to enhance its generalization across multiple plant species and environments. The application of deep learning in smart agriculture has greatly improved the accuracy of plant disease diagnosis. However, classical deep learning models, such as CNNs and ViTs, tend to be computationally intensive and are not feasible for deployment on IoT devices directly, which generally have constrained processing, memory, and power constraints. This constraint has led to the requirement of lightweight regression models that can function efficiently without reducing the sensitivity of detection in low-resource settings.

Deep learning models such as CNNs and ViTs accomplish high accuracy in plant disease detection but are computationally intensive, assembly them unsuitable for IoT devices with inadequate processing power and memory. Several IoT-based studies have explored lightweight solutions, leveraging edge computing, optimized neural networks, and sensor-based data fusion to enable tangible-time disease exposure in resource-inhibited situations. To address these challenges, a highly efficient meta-ensemble model combining MLP Mixer, LSTM, and SVM is proposed, significantly reducing parameters (~1M) and memory usage (~18.02 KB) while maintaining high classification accuracy (94.27% for Maize, 98.43% for Cotton, and 97.45% for TPP) [25]. Unlike prior IoT implementations using MobileNet, EfficientNet, and TinyML, this approach further optimizes computational efficiency with only 1.88×10^4 FLOPS, ensuring seamless deployment on microcontrollers like Raspberry Pi 4 with inference times of 0.89–2.5 seconds. The integration of edge-based processing minimizes reliance on cloud computing, thereby reducing latency and power consumption. The Level 2 SVM classifier enhances classification accuracy, surpassing conventional CNNs and ViTs while maintaining real-time performance. High AUC values (up to 0.9993) validated its reliability. This research advances IoT-driven precision agriculture by offering a scalable, low-power, and accurate plant disease detection framework tailored for real-world deployment.

Another work [26] presented an IoT-based automatic system for the classification of five varieties of leaf diseases. The platform utilizes a Raspberry Pi System on Chip (SoC) with a USB camera to take pictures of the leaves and transmit them to a host PC for analysis. An online web server allows for the real-time monitoring of images, which enables rapid and accurate disease detection. All leaf extraction is conducted on the host PC to enhance the classification quality by applying the methods of watershed and graph cutting on the collected images. Therefore, only the useful parts of the leaf were included in the extraction procedure. A Support Vector Machine (SVM) is applied in the second stage to diagnose the disease using the features extracted from GLCM. The system reached a classification accuracy of 97% on a specific dataset, which indicated that the approach used to identify diseases like Alternaria leaf disease, Bacterial Blight, Gray Mildew, Leaf Curl, and Myrothecium leaf disease was valid. More extensive testing with larger and more diverse datasets is required to determine the scalability of the system.

TCropNet model [27] is a custom convolutional neural network(CNN) which uses some pre-trained models, such as ResNet50 and EfficientNet, and it was developed with TensorFlow and Keras frameworks. The images are resized to 256 × 256 pixels, and data augmentation, such as random rotation and horizontal/vertical shifts, is performed to enhance model robustness. The AI model developed helps to identify the wheat leaves and in diagnosing the diseases with the help of object detection, which helps it achieve a whopping 99.80% classification accuracy for wheat leaf diseases. The study [28] explores shows that mobile phone CPUs and the Tesla P80 GPU have been discovered for detecting plant disease through Image analysis. The camera takes the image of plant disease, and the MobileNetV3 model is used, which is trained from Inception V3 architecture by adding separate branches to process achromatic (L channel) and chromatic (AB channels). The AI model performed ruthlessly, with a 99.54% classification accuracy on the laboratory dataset. However, the performance of the model decreased when tested on more realistic datasets, with an accuracy of 77.71%. Additionally, accuracy suffered after model quantization, particularly when using lower-precision formats such as float-16-bit and int-8-bit, with the most significant drop occurring after full int-8-bit quantization, which led to a 0.41% reduction in accuracy.

This review of existing research indicates a large need for the use of Internet of Things (IoT) technologies for Aloe Vera leaf disease detection. Although IoT-based plant disease detection is well-researched concerning crops such as tomatoes and potatoes, Aloe vera is underserved. Real-time disease monitoring in Aloe Vera using an amalgamation of IoT and deep learning models based on IoT is still very new. Previous studies mostly focused on the use of traditional image-based detection models that need to send images to the cloud for processing, which creates latency and dependence on external computing resources. Moreover, limited research has been conducted on IoT-enabled mechanisms using sensor networks, real-time analytics, and edge computing, providing a useful and economical solution for disease management in Aloe Vera farming.

This gap provides an opportunity for an IoT-based system that can address real-time environmental monitoring, deep learning-based disease classification, and edge processing to improve precision agriculture in a merged/combined framework. The proposed system leverages the edge for real-time disease diagnosis, reduces the response time, breaks free from dependency on cloud resources, and is more efficient and convenient for Aloe vera crops. Moreover, it will evaluate the consistency of the structure compared to classical methods in accuracy, efficiency, and computation time. Table 1 provides a comparative analysis of the existing literature. This systematic review serves as a foundation for advancing IoT-driven solutions for Aloe Vera disease detection**.**

3. Proposed System Architecture

An IoT-enabled Edge AI system for Aloe Vera leaf disease detection is conceptualized through various interconnected components to allow data acquisition, processing, and storage in a related general architectu

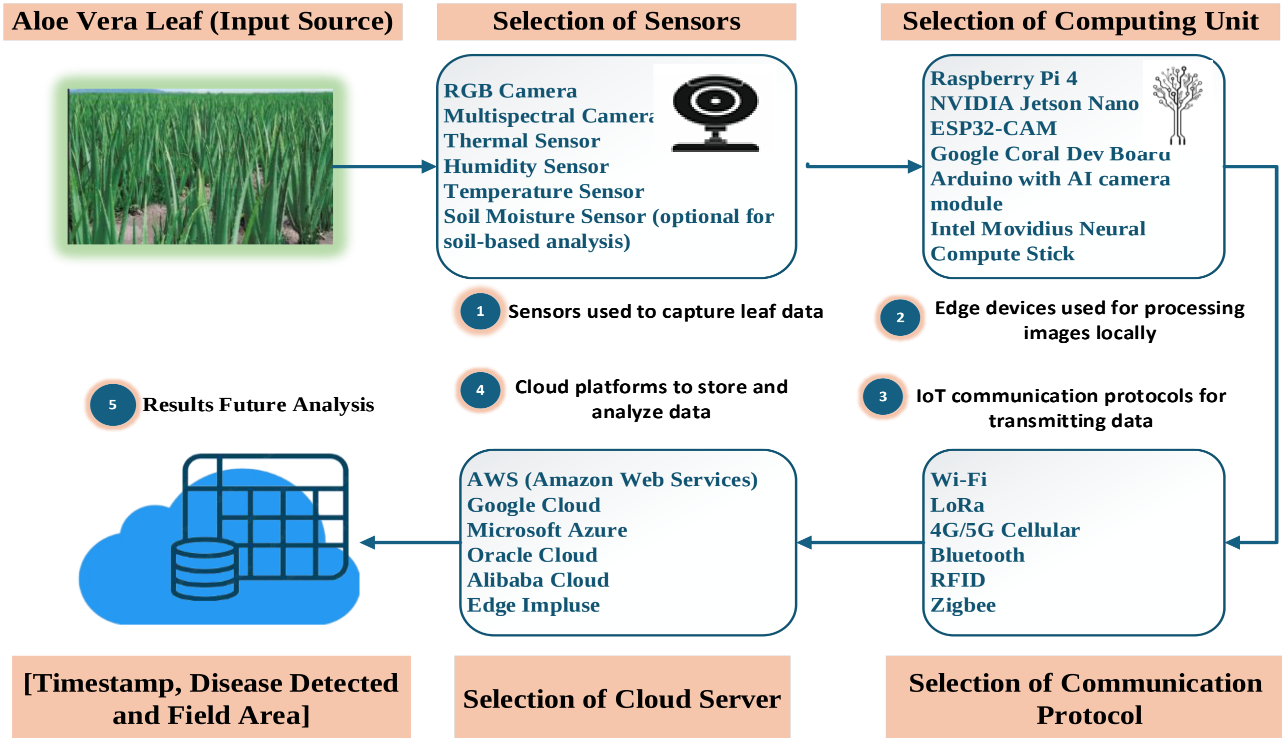


Fig. 2 General architecture

The process starts with the aloe Vera plant, specifically its leaves, which are used as the main input for detecting diseases. Different sensors are used to capture the characteristics of the leaves, such as an RGB Camera for taking a regular image and a Multispectral Camera for taking images outside the visible spectrum, which improves detection accuracy. A Thermal Sensor that measures the temperature of the surrounding air (which if not connected) can use a Humidity Sensor to identify moisture in the environment (to determine healthy plant environmental levels), a Temperature Sensor for standardizing ambient conditions, and optionally a Soil Moisture Sensor that enables soil-based analysis to supplement leaf-based analysis. It then provides local computational capabilities through various edge devices that process the captured data. For small inference tasks, we will use cost-effective devices that require minimal assembly, such as the NVIDIA Jetson Nano, Raspberry Pi 4, ESP32-CAM, Google Coral's Dev Board, Arduino with AI camera module, and Intel Movidius. Neural Compute Stick, Google Coral Dev Board, designed specifically for AI inference with edge TPU acceleration, Arduino with an AI camera module for lightweight processing and prototyping, and Intel Movidius Neural Compute Stick, which provides additional neural network acceleration for handling computationally intensive tasks. It is worth noting that any of these computational units can be employed based on the requirements of a specific application. For instance, if low power consumption and cost-effectiveness are critical, ESP32-CAM or Raspberry Pi 4 may be preferred [29]. In contrast, if high-performance deep learning inference is required, devices such as the NVIDIA Jetson Nano or Google Coral Dev Board would be more appropriate [30].

Once the data are processed at the edge, they are transmitted to cloud platforms for storage, monitoring, and further analysis through various communication protocols. These protocols include Wi-Fi, LoRa, 4G/5G Cellular, Bluetooth, RFID, and Zigbee, each offering distinct advantages in terms of range, power consumption, and data transfer rate. For instance, LoRa and 4G/5G provide long-range connectivity suitable for large farmlands, whereas Wi-Fi, Bluetooth, and Zigbee are more power-efficient but operate over shorter distances, making them ideal for localized monitoring [31]. Cloud platforms such as Microsoft Azure, Google Cloud, Oracle Cloud, AWS, Alibaba Cloud, and Edge Impulse play a crucial role in enabling continuous monitoring, historical data analysis, and predictive analysis capabilities. The choice of cloud platform can also be tailored based on factors such as accessibility, computational resources, and cost. Furthermore, this proposed system also records important details such as the identified disease, time of detection, and area of the field in the form of a spreadsheet, which is saved in the cloud (for example, Google Sheets). This systematic logging scheme makes further analysis and visualization possible, massively increasing the usefulness of the system for precision agriculture use cases. The entire structure is scalable, reusable, and energy-efficient, which makes it a building block for smart agriculture. This logging approach is represented in Figure 2, which shows how the class of the field, including the timestamp, detected disease, and field area, is systematically stored in the cloud's spreadsheet for further analysis and monitoring. Table 2 provides a comprehensive assessment of innumerable edge computing devices employed for Aloe Vera disease detection, highlighting their purpose, hardware specifications, connectivity options, transmission protocols, range, cost, and programming compatibility.

Figure 3 illustrates a comprehensive framework for an Edge AI-powered Aloe Vera Plant Disease Detection System, which comprises two major components: real-time data acquisition and application of lightweight algorithms on a Raspberry Pi device, followed by communication and data flow management. The first part of the system focuses on data acquisition, where data are gathered from two distinct sources. The first source is a sensor-driven dataset collected through Edge Impulse**,** a development platform designed for edge AI and IoT applications. Edge Impulse enables real-time data acquisition from sensors integrated with end devices situated in the field. The second source of data is an image-based pre-existing dataset provided in Kaggle, with different images related to Aloe Vera plant diseases. This dataset was used to train and improve the AI model, capturing diverse and labelled samples for both healthy and infested plants. The datasets collected from Edge Impulse and Kaggle were merged and used to create a machine-learning model. For that, this model is going to be trained using the TensorFlow model, as it is popularly known for deep learning and famous for its robustness as well as scalability. Training: An algorithm was built for accurately recognizing diseases in Aloe Vera plants based on image data and sensor readings. Once the model is trained, it undergoes a post-quantization process [32] to make it suitable for deployment on resource-inhibited devices such as the Raspberry Pi 4B. Then, by the end of the quantization process, the trained TensorFlow model is transformed into a TensorFlow Lite (TFLite) format, which creates a lightweight structure with the end goal of compatibility with edge devices. In this phase, two different versions of the model were produced. Non-quantized converted model (Float32) and quantized converted model (Int8) [33]**.** To fully benefit from an advanced model (considering size and infrastructure), the same architecture must be provided with the desired variations (quantization) while maintaining the model's state accuracy. To quickly send this model to the edge device, the same quantized model file is zipped. This file is downloaded and then transferred to the local edge device, a Raspberry Pi 4B, in the present setup. Raspberry Pi is set up to take input data from a vision-based end device like a camera/ image capturing module to enable real-time processing for the detection of diseases in the Aloe Vera plant. Cloud deployment permits the scheme to regularly monitor the health of the plant and detect diseases at an early stage anywhere, even in resource-poor environments. Comme logs show helpful information to diagnose in case of power supply uncertainty.

Table 2. Comprehensive assessment of innumerable edge computing devices

Edge Computing Device	Connectivity Transmission Protocol	Range	RAM	Price	IDE	Programming Language	Purpose
Raspberry Pi 4	Wi-Fi, 4G/5G	100m - 30 km	2-8 GB	$35-$75	Raspberry Pi OS	Python, C++	AI processing & image classification
NVIDIA Jetson Nano	Wi-Fi, 4G/5G	100m - 30 km	4 GB	$99	Ubuntu, JetPack	Python, C++	Edge AI processing, deep learning models
ESP32-CAM	Wi-Fi, Bluetooth	100m	512 KB	$5-$10	Arduino IDE	C, C++	Low-power image capturing
Google Coral Dev Board	Wi-Fi, 4G/5G	100m - 30 km	1 GB	$150	Mendel Linux	Python, C++	AI-based real-time disease detection
Arduino with AI Module	LoRa, Zigbee, RFID	1m - 15 km	2 KB	$25-$50	Arduino IDE	C, C++	Sensor data processing, IoT integration
Intel Movidius Neural Compute Stic	Wi-Fi, 4G/5G	100m - 30 km	512 MB	$79	OpenVINO Toolkit	Python, C++	AI acceleration for Edge devices

The last part of the framework includes communication and data flow management, where processed data will be stored in the cloud to make them available and analyzed later. Data are stored on the cloud, where information such as timestamps of humic acid and information about the detected disease and the area of the field are sent from the bot to the cloud. Google App Script is used to establish seamless communication between the Raspberry Pi and the cloud storage system. This script serves as a communication bridge by transmitting data to Google Sheets, where the results will then be recorded in a structured manner. This provides the necessary keys for authentication and authorization, enabling Raspberry. The data are sent from the Raspberry Pi to the Google App Script via the URL. The Google App script processes the data and updates the Google Sheet by logging the data into their respective columns. This configuration provides the availability of raw data received from the field for future analytical purposes and decision-making, which is crucial for the proposed system. Specifically, the framework focuses on the implementation of Edge AI, lightweight algorithms, and cloud-based communication systems to enable the detection of diseases in Aloe Vera plants efficiently and reliably. By deploying a lightweight model on the Raspberry Pi, this system is energy efficient and can perform real-time monitoring, which is ideal for various applications in the field with limited computational resources.

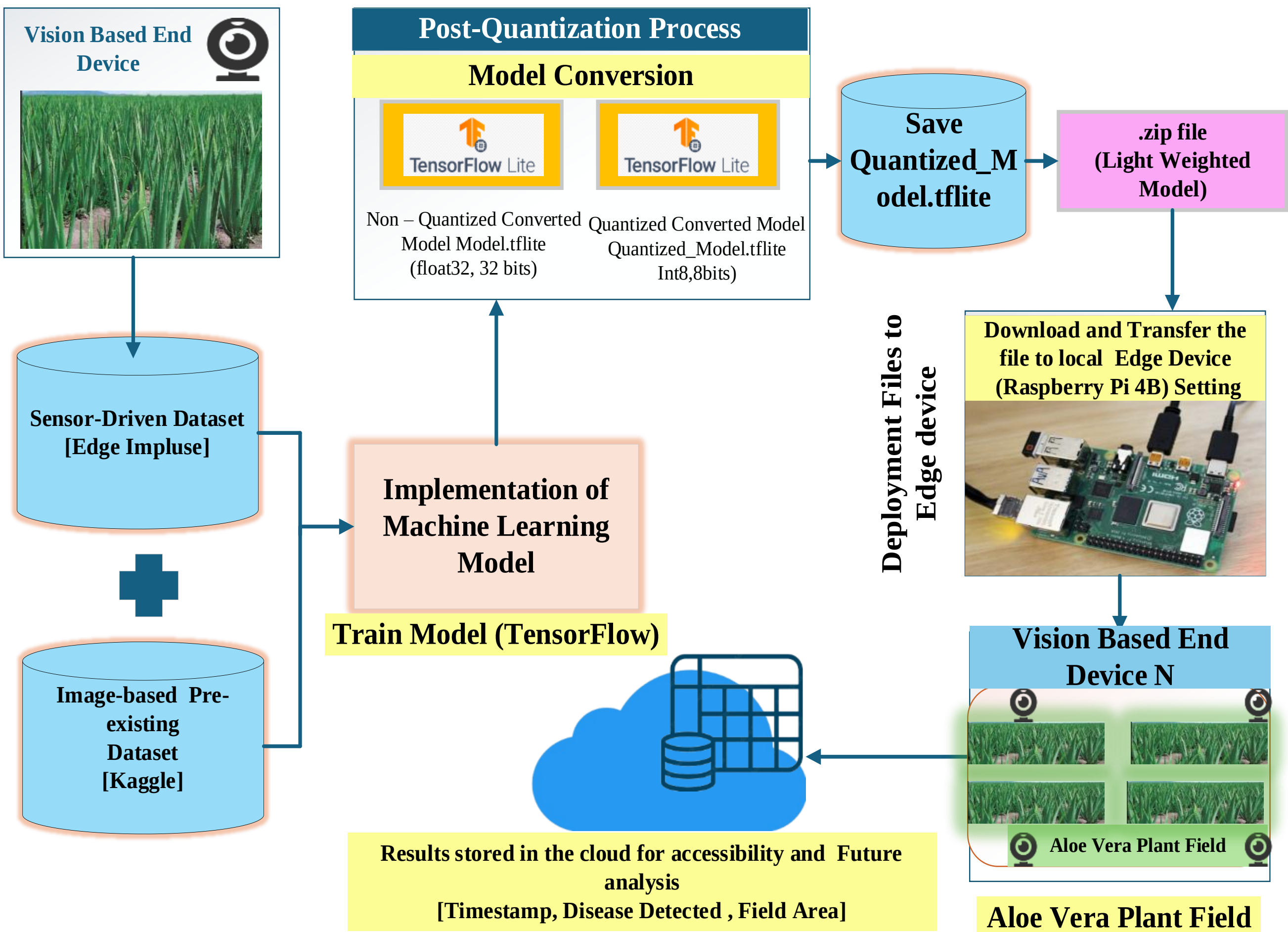


Fig. 3 Comprehensive framework for an edge AI-powered aloe vera plant disease detection system

4. Implementation of Real-Time Data Gathering and Implementation of the Lightweight Algorithm

4.1. Image Acquisition

Image acquisition takes place through an Arduino-based edge device comprising an OV7675 camera sensor to capture pictures of Aloe Vera plants, which are then sent to the Edge Impulse for dispensation. In this study, image-based data, along with sensor-driven data, is utilized to improve the accuracy of disease detection. This image dataset was obtained from Kaggle and contains 3495 high-quality JPEG images, which are divided into three classes: 1033 healthy leaf images, 1122 leaf spot images, and 1340 aloe rust images. In this way, by leveraging a labelled image dataset with real-time sensor data, the system allows for a more holistic method for Aloe Vera disease detection. To minimize overfitting and improve the generalization power [34]. The model was trained with data augmentation techniques. By using this technique, the model was exposed to more types of images, which meant that it had a better chance of generalizing to new images that never appeared in the training set because Randomization was built into the training set. The dataset was split into 80% training data, 10% validation data, and 10% test data. The validation set is utilized to optimize training, fine-tune hyperparameters, and evaluate unbiased performance, whereas the test set is completely independent of both training and validation and is used to provide an objective evaluation. Such structured partitioning decreases the risk of overfitting the model which can then be used in a production environment. Figure 4 shows the sample real-time data acquisition.

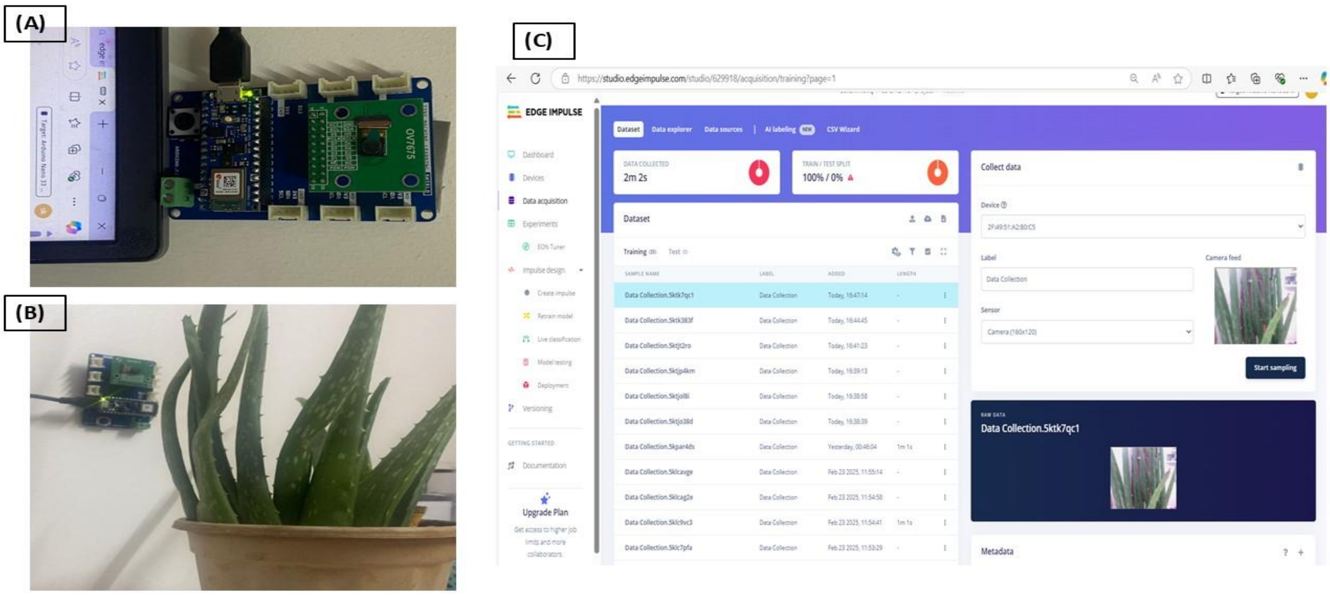


Fig. 4 Real-time data acquisition

4.2. Model Training

The proposed Aloe Vera leaf disease classification model was built on ResNet50, a deep convolutional neural network (CNN) with 50 layers, optimized for efficient feature extraction and classification. The model processed 224×244×3 resolution images to maintain uniformity across the dataset. ResNet50, pre-trained on the ImageNet dataset (which contains 1.28 million images across 1,000 classes), serves as the backbone, incorporating skip connections to mitigate vanishing gradients and enhance learning stability [35]. The architecture includes 16 residual blocks, structured as 3 blocks with 64 filters, 4 blocks with 128 filters, 6 blocks with 256 filters, and 3 blocks with 512 filters, all utilizing a 3×3 kernel size for effective feature extraction [36]. In this study, two architectures were employed: a baseline ResNet50 model and an improved Enhanced ResNet50 model specifically tailored for the task. The base ResNet50 model architecture consisted of an input layer, the ResNet50 backbone (pre-trained on ImageNet), a Global Average Pooling (GAP) layer, followed by two fully connected dense layers (512 and 256 neurons respectively), and a final dense output layer with three neurons for multi-class classification (Healthy, Leaf Spot, and Aloe Rust), comprising a total of 24,768,899 parameters, with 1,181,187 trainable parameters and 23,587,712 non-trainable parameters. To further enhance performance and generalization, the Enhanced ResNet50 model was proposed by modifying the base architecture through the addition of Batch Normalization (BN) and Dropout layers after each dense block [37]**.** This adjustment resulted in a slight increase in parameters (24,771,971 total parameters with 1,182,723 trainable parameters) but offered improved model robustness and faster convergence. The models were developed and trained in Google Colab, a cloud-based platform that provides free GPU acceleration and supports large-scale deep learning experiments efficiently. TensorFlow, an open-source deep learning framework, was utilized for model implementation, training, and evaluation, enabling flexible architecture customization and effective integration of advanced training strategies. The dataset was partitioned into 80% for training, 10% for validation, and 10% for testing, ensuring a balanced and unbiased learning process. Training for both models was conducted using mini-batch gradient descent with a batch size of 32 and the Adam optimizer, applying a learning rate decay schedule with decay steps of 1000, a decay rate of 0.96, and staircase=True, ensuring stable and efficient learning over 20 epochs. Sparse Categorical Cross-Entropy loss was utilized, and the final classification output was achieved through a softmax activation function over three classes. The inclusion of Batch Normalization layers standardized the inputs to each layer, thereby accelerating training and improving model stability, while Dropout layers introduced regularization by randomly omitting neurons during training, effectively reducing overfitting and enhancing generalization to unseen data [38].

Together, these architectural modifications made the Enhanced ResNet50 more robust and better suited for handling variability in real-world scenarios compared to the original base model. The optimized hyperparameter values are presented in Table 3, which provides configuration information on image size, number of epochs, batch size, optimizer, learning rate schedule, loss function, and kernel size.

Table 3. Hyperparameter values for proposed study

Hyperparameters	Values
Image Size	(224 * 224 * 3)
Number of Epochs	20
Number of Batches	32
Optimizer	Adam (using learning rate schedule)
Learning Rate Decay Schedule	Decay Steps: 1000 Decay Rate: 0.96 Staircase: True
Loss Function	Sparse Categorical Cross-Entropy
Kernel Size	3 × 3

4.3. Model Integration to the Raspberry Pi 4B

In this study, the Raspberry Pi 4 B was employed for edge deployment owing to its finest trade-off between cost, performance, and power proficiency. Compared to other edge devices, it provides a versatile connectivity range (Wi-Fi, 4G/5G) spanning 100m to 30 km, making it ideal for real-time plant disease monitoring in agricultural fields. Possessing between 2-8GB of RAM, you will find that even AI training models like deep learning are not a problem for this board, as is the case with lower-end machines like ESP32-CAM and Arduino, which have much less memory and performance. Also, it is being priced cheaper than higher-end competitors such as the Google Coral Dev Board ($150) and the NVIDIA Jetson Nano ($99), and also somehow manages to deliver sturdy AI processing capabilities. Due to all these reasons, Raspberry Pi 4 B is an ideal solution for scalable, low-cost, and low-power edge computing in precision agriculture. The Raspberry Pi 4 Model B is the latest version of the low-cost Raspberry Pi computer that is about the size of a credit card with the power of a desktop computer. It is a development of the previous Raspberry Pi 3 Model B, offering improved speed and functionality. Specifications It has built-in Wi-Fi, a full HDMI port, 4 USB ports and an Ethernet port It also features a display interface (DSI) and a 3.5 mm audio jack It comes with 40 GPIO pins and a camera interface (CSI) Additionally it also has a microSD card slot for storage Attached with USB HD camera that realizes 1080p, accessible to Windows or Linux (with USB 2.0), this system can provide real-time imaging of aloe vera for measurement. Software-wise, the Raspberry Pi uses the Raspbian GNU/Linux OS and runs Python scripts to implement automation and image analysis, with libraries like NumPy and OpenCV used to facilitate the analyzing of captured images in a fast and efficient manner. Figure 5 depicts the Raspberry Pi 4 B, which is a small and smart edge-computing mechanism integrated into the system.

Fig. 5 Raspberry Pi 4B

Following training, the suggested model must be used in an edge setting. Their conversion into a format that can be easily disseminated is necessary before they can be integrated with Raspberry Pi. During the training phase with TensorFlow, the proposed model underwent post-training quantization to optimize its performance for deployment on edge devices [39]. This process involves converting the trained TensorFlow (Keras) model into the TensorFlow Lite (TFLite) format using the TensorFlow Lite Converter (TFLiteConverter) [40]**.** The conversion is crucial for enabling the model to run efficiently on resource-inhibited devices**,** such as microcontrollers, mobile phones, and Raspberry Pi**.** As part of this transformation, quantization was pragmatic to diminish the model size and boost its computational efficiency. This is achieved by lowering the correctness of model weights from 32-bit floating point to 8-bit integers [41], assembly is predominantly advantageous for edge, where power and dispersion capabilities are limited. A TFLite model is a self-contained file that encapsulates both the floating-point and quantized parameters**,** including optimized weight and bias values**.** This format allows the model to perform inference independently without requiring the full TensorFlow framework**.** Additionally, quantized models offer faster inference speeds, tumbling latency, and power consumption, which are decisive for real-time applications on edge devices. Moreover, the smaller model **size** enables efficient storage and deployment, making it ideal for IoT and embedded AI solutions.

5. Results and Discussion

The Results and Discussion section presents the key findings of the study, supported by relevant data, figures, and tables. This section presents the evaluation metrics and detailed analysis of model performance. It further discusses the efficiency of the proposed approach in Edge AI environments and its effectiveness when applied to real-world scenarios.

5.1. Performance Evaluation Metrics

The performance of the projected model was assessed using numerous metrics, including accuracy, precision, recall, and F1-score. These metrics are consequent from the confusion matrix, which comprises true-positive (T.P.), true-negative (T.N.), false-positive (F.P.), and false-negative (F.N.) predictions. In this study, the classification task involves three classes: Aloe Rust, Leaf Spot, and Healthy Leaf. The following mathematical formulas use TF, TN, FP, and FN to calculate accuracy, Precision m, and recall F1 measure:

$$Accuracy = \frac{True\ Positive + \text{True Positive + Trueegative}}{True\ Positive + True\ Negative + False\ Positive + False\ Nagative} \quad (1)$$

The percentage of correctly predicted positives to all positive predictions for a particular class can be used to define the precision in divergence(2).

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

Recall is a dimension assigned to the portion of all positive trials that are correctly expected to be positive. It is computed by(3)

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Nagative} \quad (3)$$

For the F1-score, the weighted mean of Precision and Recall is used, as shown in (4):

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad 4)$$

5.2. Model Performance Analysis

Table 4 compares the performance of ResNet50 with and without batch normalization and dropout layers**,** showing improvements in model accuracy and loss metrics. The baseline ResNet50 attained a training accuracy of 99.40% with a low training loss of 0.0093; however, its validation accuracy (96.88%) and testing accuracy (97.08%) were slightly lower, indicating potential overfitting. In contrast, the enhanced ResNet50 with batch normalization and dropout improved the training accuracy to 99.57%**,** reduced the training loss to 0.0090, and significantly boosted the validation accuracy to 98.30% while lowering the validation loss to 0.0595**.** The most notable improvement was observed in testing accuracy (99.15%) and testing loss (0.0508)**,** confirming that adding batch normalization and dropout layers effectively mitigates overfitting, enhances generalization, and improves overall model robustness for real-world applications. The Baseline ResNet50 model demonstrates rapid convergence, with training accuracy reaching nearly 99% within the first 5 epochs**,** but the validation accuracy fluctuates significantly between 1 to 20 epochs, indicating overfitting. The training loss decreased steadily, whereas the validation loss remained inconsistent, suggesting poor generalization.

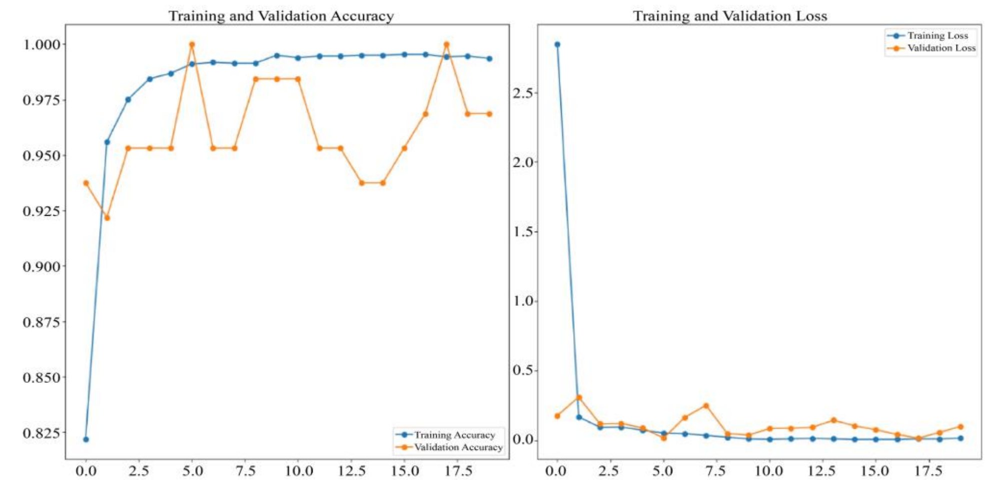
In contrast, the Improved ResNet50 model, which incorporates batch normalization and dropout layers, shows a more stable training process across 1 to 20 epochs. While the validation accuracy exhibits some oscillations, it follows a more consistent upward trend, and the validation loss remains more controlled.

This suggests that additional layers help mitigate overfitting and enhance the aptitude of the model to generalize efficiently across unseen data. Figure 6 illustrates a comparison of the training and validation performance between the baseline and improved ResNet50 models.

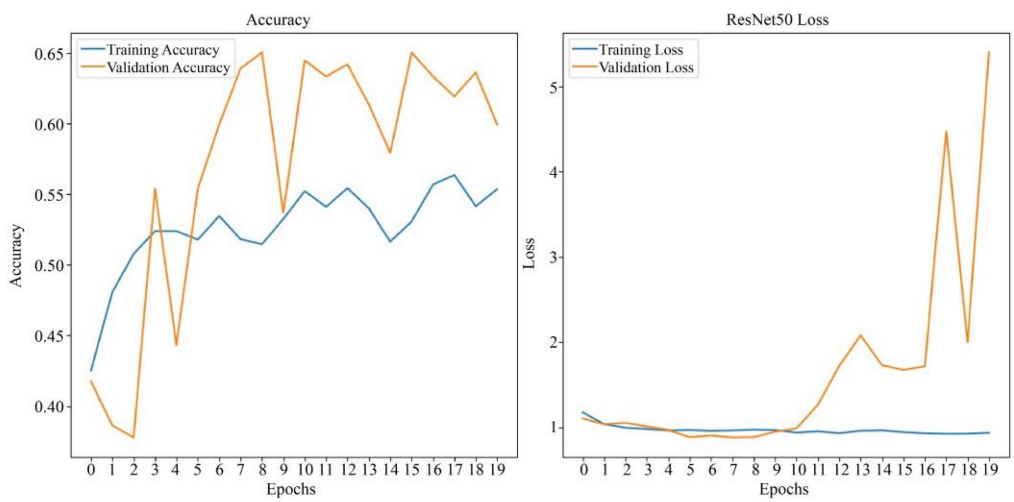
Table 4. Performance metrics of different models including training, validation, and testing accuracy and loss

Models	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Testing Accuracy	Testing Loss
Resnet50	0.9940	0.0093	0.9688	0.1011	0.9708	0.1659
Resnet50 + Batch layer + Dropout layer	0.9957	0.0090	0.9830	0.0595	0.9915	0.0508

The performance evaluation of the Improved ResNet50 model, incorporating batch normalization and dropout layers**,** demonstrates significant improvements over the baseline model in classification accuracy, as reflected in both confusion matrices and statistical performance metrics. The confusion matrices in Figure 7 reveal that while the Baseline ResNet50 model misclassified several images particularly 5 Leaf Spot images as Aloe Rust and 3 Healthy Leaf images as Leaf Spot the Improved ResNet50 model significantly reduced these error, achieving near-perfect classification with only 1 misclassification in Aloe Rust and 2 in Leaf Spot, while correctly classifying all Healthy Leaf images. In terms of statistical performance, the improved model achieved a precision of 0.9920, recall of 0.9921**,** and an F1-score of 0.9920, significantly outperforming the baseline model's precision of 0.9708, recall of 0.9688, and F1-score of 0.9698**,** with an overall accuracy improvement from 97.08% to 99.15%, as shown in Table 5. Over 20 training epochs, the improved model demonstrated faster convergence, lower validation loss, and higher validation accuracy**,** highlighting its ability to generalize better while avoiding overfitting, which was more prominent in the baseline model. These enhancements, driven by batch normalization and dropout layers**,** result in a more stable and robust model capable of delivering highly reliable Aloe Vera disease classification, making it well-suited for real-world agricultural applications.

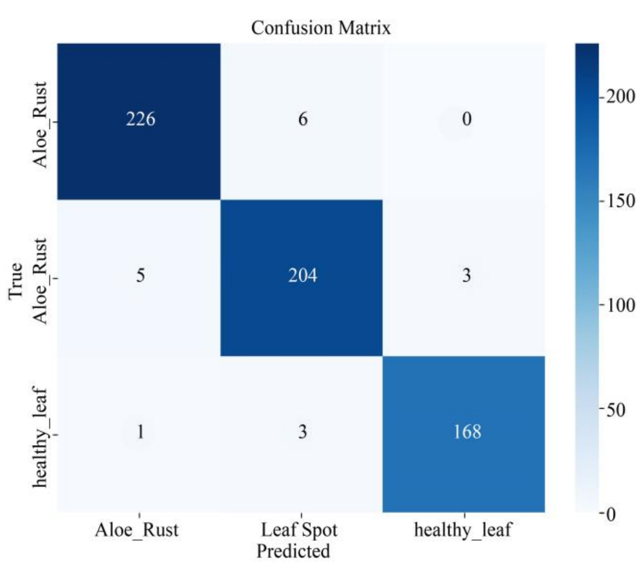


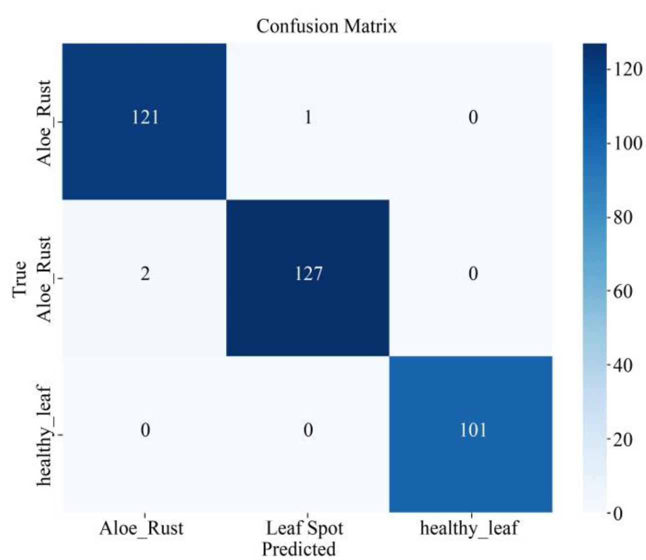
(a) Base Resnet50 model



(b) Improved ResNet50 model

Fig. 6 Comparison of training and validation performance between baseline and improved ResNet50





(a) Base Resnet50 mode **(b) Improved ResNet50 model**

Fig. 7 Confusion matrix (a) Baseline ResNet50, and (b) Improved ResNet50.

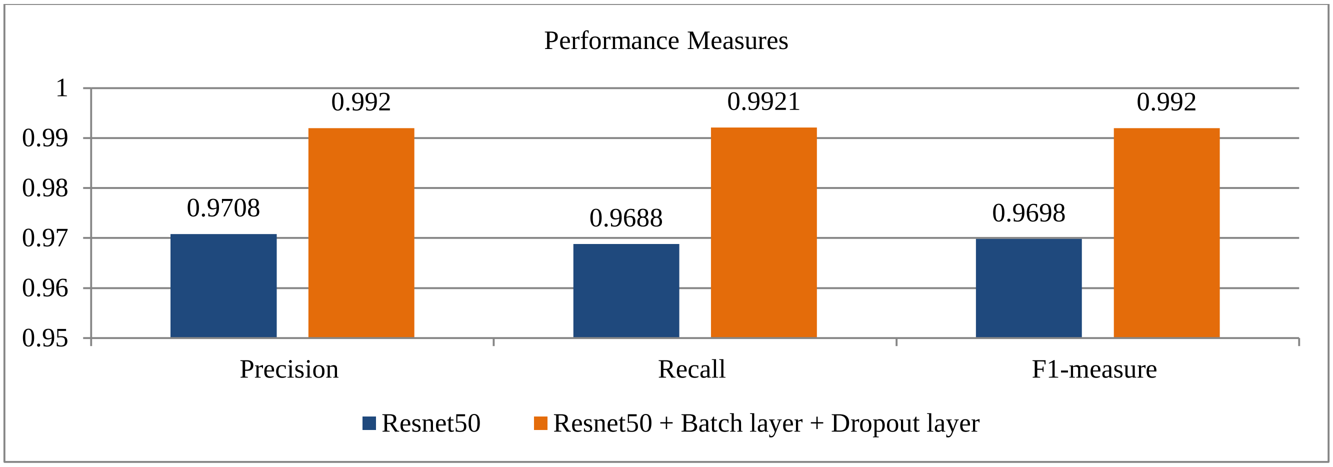


Fig. 8 Performance measure between baseline and improved ResNet50

Table 5. Performance measures

Models	Precision	Recall	F1 - measure
ResNet50	0.9708	0.9688	0.9698
ResNet50 + Batch layer + Dropout layer	0.9920	0.9921	0.9920

5.3. Edge AI Performance

Microcontrollers (MCUs) and microprocessors (MPUs) are widely used for positioning deep learning models on edge devices, but they differ significantly in computational power and memory utilization. MCUs, designed for low-power applications, exhibit high latency and limited RAM (2.5MB–2.7MB), making them suitable only for small-scale AI tasks, whereas MPUs like the CPU and GPU on Raspberry Pi 4 provide significantly higher computational power and memory, enabling efficient deep learning inference. The performance evaluation of the quantized ResNet50 model (INT8, 23.4MB) on different edge devices highlights the significant differences between microcontrollers (MCUs) and microprocessors (MPUs) in terms of latency, RAM, and ROM usage. MCUs, designed for low-power applications, exhibit extremely high latency, with low-end MCUs taking ~2.88 hours (10.3 million ms) for inference, while high-end MCUs with AI accelerators reduce latency to ~133 seconds but remain slow. In contrast, MPUs like the Raspberry Pi 4 CPU process inference in just ~4.9 seconds, and a GPU/AI accelerator further improves it to 0.8 seconds, demonstrating the efficiency of microprocessors in real-time tasks. Memory constraints are another key challenge—MCUs have strict RAM limitations (2.5MB–2.7MB), restricting their capability to run deep learning models, whereas MPUs have significantly higher RAM availability**,** enabling smoother execution. The original ResNet50 model (FP32) was 102.4MB, which is impractical for edge deployment, but quantization (INT8) reduced it to 23.4MB**,** allowing it to fit within embedded systems' memory constraints with minimal precision loss. Although MCUs struggle with deep learning workloads, MPUs, especially with GPU acceleration, provide a practical solution for deploying AI models on edge devices in real-time applications, making them ideal for Aloe Vera disease classification and other agriculture-based edge AI tasks**.**

Table 6. Performance comparison of model deployment on different devices, including raspberry Pi 4

Device	Latency (ms)	RAM	ROM	Quantized Model Size (INT8)	Original Size Model (FP32)
Low-end MCU	10,373,618 ms (~2.88 hrs)	2.5MB	23.2MB	23.4MB	102.4 MB
High-end MCU	133,472 ms (~133 sec)	2.7MB	23.2MB	23.4MB	102.4 MB
High-end MCU + AI Accelerator	133,472 ms (~133 sec)	2.7MB	23.2MB	23.4MB	102.4 MB
CPU (Raspberry Pi 4)	4,922 ms (~4.9 sec)	Higher	23.4MB	23.4MB	102.4 MB
GPU / AI Accelerator	821 ms (~0.8 sec)	Higher	23.4MB	23.4MB	102.4 MB

5.4. Effectiveness in Real-World Scenarios

Figure 9 presents examples of accurately classified images from the test dataset, displaying both the predicted and actual classes along with their respective confidence scores. The results indicated that the model successfully identified the class of each image with 100% confidence in all cases. The first image corresponds to the Leaf Spot, while the second and third images are categorized as Healthy Leaf. The latency values for each classification were also recorded, with times of 0.0867s, 0.0667s, and 0.0663s, respectively. Predictions with higher confidence values suggest that the model is highly certain regarding its classification, reducing the need for additional verification. The ability to make quick and accurate predictions is crucial in real-time applications, such as plant disease monitoring, where immediate decision-making can significantly impact agricultural productivity. These outcomes determine the efficacy of the projected edge-based deep learning model in accurately detecting plant health conditions with minimal latency.

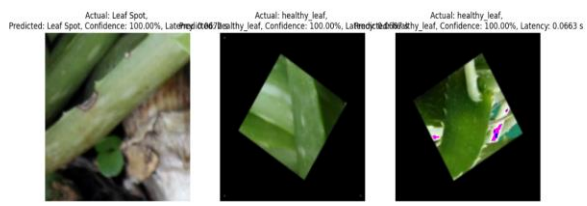
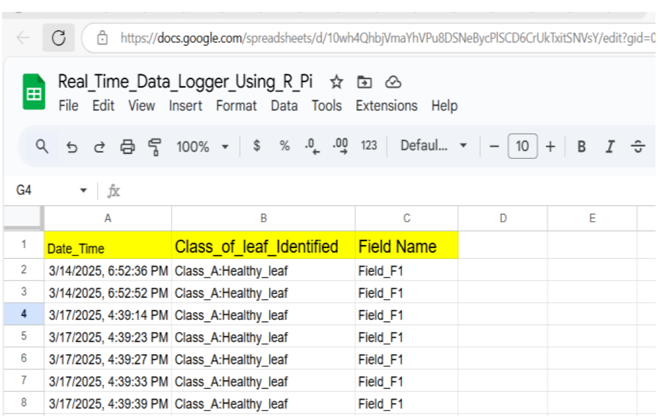


Fig. 9 Result



	A	B	C
1	Date_Time	Class_of_leaf_Identified	Field Name
2	3/14/2025, 6:52:36 PM	Class_A:Healthy_leaf	Field_F1
3	3/14/2025, 6:52:52 PM	Class_A:Healthy_leaf	Field_F1
4	3/17/2025, 4:39:14 PM	Class_A:Healthy_leaf	Field_F1
5	3/17/2025, 4:39:23 PM	Class_A:Healthy_leaf	Field_F1
6	3/17/2025, 4:39:27 PM	Class_A:Healthy_leaf	Field_F1
7	3/17/2025, 4:39:33 PM	Class_A:Healthy_leaf	Field_F1
8	3/17/2025, 4:39:39 PM	Class_A:Healthy_leaf	Field_F1

Fig. 10 Logged data from google sheet

The Google Sheet shown in Figure 10 in the image represents real-time logging of leaf classification results transmitted from the Raspberry Pi using Google App Script. The Date_Time column records the timestamp of each entry, while the Class_of_leaf_Identified column logs the detected leaf class, which in this case is consistently "Class_A: Healthy_leaf." The Field Name column categorizes the data under Field_F1**,** possibly indicating a specific sensor or dataset classification. The Raspberry Pi captures leaf images, processes them using a deep learning model, and securely transmits the classification results to Google Sheets via a pre-configured Google App Script URL with authentication keys. This setup ensures structured, real-time data collection for efficient plant health monitoring, enabling future analysis and informed decision-making in agricultural applications.

6. Conclusion

The proposed Aloe Vera leaf disease classification model, based on ResNet50, achieved high classification performance with an accuracy of 99.15%, precision of 99.20%, recall of 99.21%, and an F1-score of 99.20%. The integration of batch normalization and dropout layers effectively reduces overfitting, thereby improving the generalization capability of the model. The deployment of the quantized TFLite model on Raspberry Pi 4 B ensures efficient edge computing, enabling real-time disease detection with reduced latency and computational overhead. The model achieves an inference latency of 4,922 ms (~4.9s) on Raspberry Pi 4, utilizing higher RAM with a quantized model size of 23.4MB, significantly reducing memory usage from 102.4MB (FP32) to 23.4MB (INT8) while maintaining classification accuracy. These advancements have contributed to the growing field of AI-driven precision farming, offering practical and scalable solutions for farmers. These conclusions highlight the potential of edge-based deep learning for scalable and cost-effective plant disease detection, paving the way for further advancements in smart precision farming. Future work can focus on hardware optimization by integrating AI accelerators or low-power GPUs to improve inference speed, expanding the dataset to enhance model robustness across diverse environmental conditions, and developing a hybrid cloud-edge architecture to balance latency, computational efficiency, and scalability in agricultural applications.

References

[1] Giuseppe Cristiano, Bernardo Murillo-Amador, and Barbara De Lucia, "Propagation Techniques and Agronomic Requirements for the Cultivation of Barbados Aloe (Aloe vera (L.) Burm. F.)-A Review," *Frontiers Research Foundation*, vol. 7, pp. 1-14, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[2] Nadia Mohamed Said Arafa, Huda Mohammad Ahmad Hummadi, and Gehan Moustafa Badr, "The Potential of Aloe Vera Gel Utilization for Skin Wound Healing in Rats based on GC–MS and HPLC Chemical Profile," *The Journal of Basic and Applied Zoology*, vol. 86, pp. 1-15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] A. Catalano et al., "Aloe Vera-An Extensive Review Focused on Recent Studies," *Foods*, vol. 13, no. 13, pp. 1-55, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Aloe Vera Extract Market Size Statement, 2019-2025, Grand View Research, pp. 1-90, 2025. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/aloe-vera-extracts-market

[5] Aloe Vera Market Size and Share Outlook - Forecast Trends and Growth Analysis Report (2025-2034), Expert Market Research, 2025. [Online]. https://www.expertmarketresearch.com/reports/aloe-vera-market

[6] Olwen M. Grace et al., "Evolutionary History and Leaf Succulence as Explanations for Medicinal Use in Aloes and the Global Popularity of Aloe Vera," *BMC Evolutionary Biology*, vol. 15, pp. 1-12, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Xihai Zhang, Zhanyuan Cao, and Wenbin Dong, "Overview of Edge Computing in the Agricultural Internet of Things: Key Technologies, Applications, Challenges," *IEEE Access*, vol. 8, pp. 141748-141761, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[8] Prem Rajak et al., "Internet of Things and Smart Sensors in Agriculture: Scopes and Challenges," *Journal of Agriculture and Food Research*, vol. 14, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Sebastian Sadowski, and Petros Spachos, "Wireless Technologies for Smart Agricultural Monitoring using Internet of Things Devices with Energy Harvesting Capabilities," *Computers and Electronics in Agriculture*, vol. 172, pp. 1-11, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Fazeel Ahmed Khan, Adamu Abubakar Ibrahim, and Akram M. Zeki, "Environmental Monitoring and Disease Detection of Plants in Smart Greenhouse using Internet of Things," *Journal of Physics Communications*, vol. 4, no. 5, pp. 1-15, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Rubina Rashid et al., "An Early and Smart Detection of Corn Plant Leaf Diseases Using IoT and Deep Learning Multi-Models," *IEEE Access*, vol. 12, pp. 23149-23162, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Zhiyan Liu et al., "Internet of Things (IoT) and Machine Learning Model of Plant Disease Prediction-Blister Blight for Tea Plant," *IEEE Access*, vol. 10, pp. 44934-44944, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[13] Monalisa Mishra, Prasenjit Choudhury, and Bibudhendu Pati, "Modified Ride-NN Optimizer for the IoT based Plant Disease Detection," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 691-703, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Geetam Tomar, "Welcome from CSNT 2021 General Chair Proceedings," *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, Bhopal, India, pp. 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Munira Akter Lata et al., "Development of an IoT based Smart Potato Leaf Diseases Monitoring and Controlling System with Image Processing," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 22, no. 6, pp. 1502-1510, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Zinon Zinonos et al., "Grape Leaf Diseases Identification System Using Convolutional Neural Networks and LoRa Technology," *IEEE Access*, vol. 10, pp. 122-133, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Juan Contreras-Castillo et al., "SAgric-IoT: An IoT-Based Platform and Deep Learning for Greenhouse Monitoring," *Applied Sciences*, vol. 13, no. 3, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] S. Aasha Nandhini et al., "Web Enabled Plant Disease Detection System for Agricultural Applications Using WMSN," *Wireless Personal Communications*, vol. 102, pp. 725-740, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[19] Sehan Kim et al., "IoT-Based Strawberry Disease Prediction System for Smart Farming," *Sensors*, vol. 18, no. 11 pp. 1-17, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ninhi Kundu et al., "IoT and Interpretable Machine Learning based Framework for Disease Prediction in Pearl Millet," *Sensors*, vol. 21, no. 16, pp. 1-23, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] S.V.S. Ramakrishnam Raju et al., "Design and Implementation of Smart Hydroponics Farming Using IoT-Based AI Controller with Mobile Application System," *Journal of Nanomaterials*, vol. 2022, no. 1, pp. 1-12, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Aminu Musa et al., "Low-Power Deep Learning Model for Plant Disease Detection for Smart-Hydroponics Using Knowledge Distillation Techniques," *Journal of Low Power Electronics and Applications*, vol. 12, no. 2, pp. 1-20, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Sushruta Mishra et al., "A Smart and Sustainable Framework for Millet Crop Monitoring Equipped with Disease Detection using Enhanced Predictive Intelligence," *Alexandria Engineering Journal*, vol. 83, pp. 298-306, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Nermeen Gamal Rezk et al., "An Efficient Plant Disease Recognition System Using Hybrid Convolutional Neural Networks (CNNs) and Conditional Random Fields (CRFs) for Smart IoT Applications in Agriculture," *International Journal of Computational Intelligence Systems*, vol. 15, pp. 1-21, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ritesh Maurya, Satyajit Mahapatra, and Lucky Rajput, "A Lightweight Meta-Ensemble Approach for Plant Disease Detection Suitable for IoT-Based Environments," *IEEE Access*, vol. 12, pp. 28096-28108, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[26] Hiren K. Mewada, and Jignesh J. Patoliya, "IoT based Automated Plant Disease Classification using Support Vector Machine," *International Journal of Electronics and Telecommunications*, vol. 67, no. 3, pp. 517-522, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Oumayma Jouini et al., "Wheat Leaf Disease Detection: A Lightweight Approach with Shallow CNN Based Feature Refinement," *Agri Engineering*, vol. 6, no. 3, pp. 2001-2022, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[28] Hoang Trong Minh, Tuan Pham Anh, and Van Nguyen Nhan, “A Novel Light-Weight DCNN Model for Classifying Plant Diseases on Internet of Things Edge Devices,” *MENDEL-Soft Computing Journal*, vol. 28, no. 2, pp. 2571-3701, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Asad Ul Haq Hashmi et al., “Effects of IoT Communication Protocols for Precision Agriculture in Outdoor Environments,” *IEEE Access*, vol. 12, pp. 46410-46421, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[30] Stephan Patrick Baller et al., “DeepEdgeBench: Benchmarking Deep Neural Networks on Edge Devices,” *2021 IEEE International Conference on Cloud Engineering (IC2E)*, San Francisco, CA, USA, pp. 20-30, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[31] Lameya Aldhaheri et al., “LoRa Communication for Agriculture 4.0: Opportunities, Challenges, and Future Directions,” *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 1380-1407, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[32] Carlos Victorino Padeiro et al., “Lightweight Maize Disease Detection through Post-Training Quantization with Similarity Preservation,” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA, pp. 2111-2120, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[33] Pierre-Emmanuel Novac, “Quantization and Deployment of Deep Neural Networks on Microcontrollers,” *Sensors*, vol. 21, no. 9, pp. 1-32, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[34] Connor Shorten, and Taghi M. Khoshgoftaar, “A Survey on Image Data Augmentation for Deep Learning,” *Journal of Big Data*, vol. 6, pp. 1-48, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[35] S. Jayashree, and V. Sumalatha, “Plant Leaf Disease Detection Using Resnet-50 Based on Deep Learning,” *Proceedings of the International Conference on Digital Transformation in Business: Navigating the New Frontiers Beyond Boundaries*, pp. 150-166, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[36] Jin Liang, and Wenping Jiang, “A ResNet50-DPA Model for Tomato Leaf Disease Identification,” *Frontiers in Plant Science*, vol. 14, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[37] Mattia Segu, Alessio Tonioni, and Federico Tombari, “Batch Normalization Embeddings for Deep Domain Generalization,” *Pattern Recognition*, vol. 135, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Imrus Salehin, and Dae-Ki Kang, “A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain,” *Electronics*, vol. 12, no. 14, pp. 1-23, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[39] Oumayma Jouini et al., “A Survey of Machine Learning in Edge Computing: Techniques, Frameworks, Applications, Issues, and Research Directions,” *Technologies*, vol. 12, no. 6, pp. 1-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[40] Konkala Venkateswarlu Reddy et al., “Edge AI in Sustainable Farming: Deep Learning-Driven IoT Framework to Safeguard Crops from Wildlife Threats,” *IEEE Access*, vol. 12, pp. 77707-77723, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[41] Dušan Markovic et al., “Image Processing for Smart Agriculture Applications Using Cloud-Fog Computing,” *Sensors*, vol. 24, no. 18, pp. 1-26, 2024. [CrossRef] [Google Scholar] [Publisher Link]

Appendix

Table 1. Summary of Previous Study

Reference	Hardware	Sensors	Communication Protocol	Data Transmission Optimization	Cloud Platform	Edge Processing	Software Used / Platform	Functionality &AI Model	Result	Limitation
[10]	Microcontrollers (Arduino), VGA camera module (OV7670),	DHT-22 (Temperature & Humidity), Moisture Sensor, Rain Sensor	Wireless Communication	IoT-based hardware prototype for environmental data collection	Not specified	NA	MATLAB	Image pre-processing, Image segmentation, K-Means clustering algorithm, Feature extraction, Alex Netoptimized CNN architecture	NA	Utilizing cloud AI models for remote disease classification
[11]	Arduino-based Smart Node Hub (SNH) with sensors and Bluetooth module	Soil moisture, temperature, atmospheric pressure, and humidity sensors	Bluetooth for data transmission to external devices	PLX-DAQ software for real-time data logging and transmission to Excel (CSV)	Cloud-based computing for scalable disease detection and monitoring	Edge computing for real-time environmental data analysis and disease alerts	PLX-DAQ, Arduino IDE, sklearn	MMF-Net: A Convolutional Neural Network (CNN) based multi-model fusion architecture that fuses multiple contextual features (RL-block, PL-blocks) for accurate plant disease classification.	Achieved 99.23% accuracy	Limited environmental factors, no real-time edge computing, and internet dependency

[12]	Arduino-based IoT prototype	DHT-22 (Temperature & Humidity), Rain Sensor	NA	NA	NA	NA	Scikit-Learn (Python), Regression Models	NA	NA	Limited environmental factors considered; no real-time internet or edge computing integration
[13]	IoT nodes, (Cameras) Sink node	High-specification cameras nodes to sink node	RPL Protocol (IoT-nodes to sink Node)	Image collection and transmission through IoT	Not mentioned	Yes (sink node processes images)	SCA-based RideNN classifier	Median filter for image pre-processing, segmentation, and feature extraction, SCA-based RideNN, optimizing neural network weights.	Accuracy: 91.56%, Sensitivity: 94.04%, Specificity: 92.98%, Energy efficiency: 0.1734	Detects only diseased vs. healthy plants, does not classify disease type
[14]	IoT Nodes (Cameras, MY THINGS Smart Sensor, Robotic Arm, Arduino Uno)	Proximal Soil Sensor (PSS), Temperature Sensors, Water Quality Sensors, GPS	Wifi	Automated data collection	Not mentioned	Yes (image processing & decision-making)	NA	Local Binary Thresholding, Genetic Algorithm for Image Recognition	NA	Does not classify disease type, requires farmer input for robotic arm control.
[15]	Arduino UNO	ESP32-CAM, DHT22 Temperature and Humidity Sensor, GSM/GPRS Module (SIM900A)	GSM/GPRS (SIM900A) for sending SMS alerts	Not mentioned	Arduino UNO processes the data from sensors and image data. Local	Arduino IDE: Used for programming the Arduino UNO.	NA	ResNet-50: leaf disease classification Temperature and Humidity Monitoring: The	97% accuracy in recognizing early and late blight diseases 97% accuracy in recognizing early and late blight	May require internet connectivity for real-time feedback and more accurate monitoring

		:			detection of diseases (early blight, late blight) is done using the ResNet-50 model.			system tracks environmental parameters and alerts farmers when conditions	diseases	.
[16]	LoRa-enabled IoT nodes	Camera (low-resolution grayscale)	LoRa	Image size reduction, grayscale transformation, packet loss handling	Not mentioned	NA	Not mentioned	CNN, Grad-CAM (XAI)		Limited bandwidth, low duty cycle (1%), reduced image quality
[17]	Microchip PIC18LF46 K22, Raspberry Pi 3 Model BRaspberry Pi 3 Model B	ESP32-CAM module, temperature, humidity, soil moisture, pH sensors,	ZigBee (sensor nodes), Wi-Fi (camera nodes), Cellular (gateway node to cloud)	Low-power D2D communication, optimized radio technologies for efficient networking, multi-hop communication for extended range	Cloud-based processing and storage, accessible via web and mobile applications	Gateway node processes images using a trained CNN model for real-time disease detection before transmitting results to the cloud	IoT platform (SAgric-IoT), Deep Learning (CNN for disease detection), web and mobile applications	CNN model	95%	potential connectivity issues in remote areas
[18]	Raspberry Pi 3 with built-in Wi-Fi	Wireless Multimedia Sensor Networks (WMSN) with camera capability	Wi-Fi (via Raspberry Pi 3)	Compressed Sensing (CS) to reduce data overhead	ThingSpeak	Raspberry Pi 3 for initial image processing before cloud transmission	OpenCV (for image processing), Python (for implementation)/ MATLAB	Image capture, segmentation, feature extraction, classification, and cloud-based	NA	Needs real-field deployment and testing for practical validation

								monitoring		
[20]	Drone cameras, digital cameras, and Raspberry Pi	Environmental sensors (pH, temperature, humidity, etc.), Imaging sensors	Wi-Fi MQTT	Transfers only relevant data to the cloud - Stores up to 100 images on Raspberry Pi before offloading to the cloud	AWS Cloud (Amazon Web Services)	Yes, using Raspberry Pi	TensorFlow/Keras - Python - Custom-Net Model - Grad-CAM for feature visualization	Custom-Net deep learning mode	98.78% classification accuracy, VGG-19 achieved 99.39% accuracy but required higher training time	High training time for VGG-19, Custom-Net has lower accuracy than VGG-19, Higher computational cost for deep models
[21]	Raspberry Pi, IoT Sensors, Actuators, Camera Module	NPK, pH, Water Level, Turbidity, Temperature, Humidity, Sunlight, Camera	IoT Cloud, Wireless Communication	Real-time sensor data transmission via IoT cloud, optimized by Raspberry Pi processing	IoT Cloud Server	es, real-time edge processing via Raspberry Pi	Agri-Hydroponic Android App, Deep Learning Convolutional Neural Network (DL-CNN)	Prediction-DLCNN for nutrient level estimation, Classification-DLCNN for disease detection	Accuracy: 99.29%, F-Measure: 99.23%; Automated nutrient supply and disease control	diverse environmental conditions
[22]	Raspberry Pi	Cameras	MTT	Energy harvesting for edge device power supply - Offline operation for reduced energy consumption	None	Yes, using Raspberry Pi	NA	Low-power CNN model for plant disease detection in smart hydroponics - Knowledge Distillation to reduce model size and power consumption	9.4% accuracy - Power consumption: 6.22W (30% reduction) - 2.4% accuracy improvement compared to previous models	Limited to diseases trained on - Offline model, no network connectivity for updates or additional processing
	Raspberry	Temperatur	NA	NA	Cloud	Raspberry	PyTorch	Customize	Accuracy:	Drone

[23]	Pi, IoT Sensors**,** NVIDIA GeForce GTX 1650ti GPU	e Sensors, Humidity Sensors, Soil Moisture Sensors			server	Pi performs local processing and disease detection using the Customize d-CNN model. Local alerts are triggered if abnormal sensor readings are detected.	1.9.1: Deep learning framework for model development. - CUDA 12.1 and cuDNN 8.9.0: Utilized for GPU acceleration during model training.	d-CNN Model	98.8% - Precision: 98.2% - Recall: 97.4% F-score: 97.7%. The system performed efficiently with minimal training and testing delays (67s and 88s, respectively). Customized-CNN Model	services for better precision and extending the system to mobile platforms
[24]	IoT-Hub, IoT Devices, Actuators, Controllers.	Temperature, Humidity, CO_2, Illumination.	RS-485, CAN, LoRa, oneM2M, IPSO, HTTP	LoRa-based wireless transmission, cloud-based processing.	FaaS	Yes, sensor data is processed at the edge before cloud upload.	OneM2M-based IoT platform.	General Infection Model (GIM) predicts infection risks based on temperature and wetting duration.	Infection probabilities below 0.8 are managed via ventilation, humidity control, and plant removal; above 0.8 requires chemical treatment	Adoption of AI models to varying environmental conditions.
[25]	Raspberry Pi 4 Model B (4GB RAM)	Image Sensors (RGB)	Wi-Fi, MQTT	Optimized data compression		Yes (MLP Mixer + LSTM + SVM on-device)	Python, TensorFlow, Scikit-learn	Lightweight meta-ensemble for plant disease detection using MLP Mixer, LSTM, and SVM	Overall Accuracy: 96.72%Inference time: 0.89–2.5s, Memory: 18.02KB, FLOPS: 1.88 × 10^4	Limited to small-scale IoT devices, may need retraining for new disease types
[26]	Raspberry Pi SoC	USB Camera	NOT Mentioned	Images processed	Online Web	Image acquisition	MATLAB	Full leaf extraction	97% classification	Requires further

				on the host PC after transmission from the Raspberry Pi	Server	and classification on the host PC		using watershed & graph cut - Disease classification using SVM with features from GLCM	accuracy	testing with larger datasets
[28]	IoT-based system, Nvidia Jetson Nano	Cameras	Various IoT protocols (e.g., Wi-Fi, Zigbee)	Optimization through edge processing (minimal data transfer)	Not specified	Real-time on Nvidia Jetson Nano	TensorFlow, Keras (custom CNN, pretrained models like ResNet50, EfficientNet)	CropNet for disease detection with data preprocessing (256×256 pixels) and data augmentation (random rotation, shifts, etc.)	Achieved 99.80% classification accuracy	Performance may decrease in distinguishing closely related diseases; dataset limitations may affect real-world application accuracy.
[29]	Mobile phone CPUs, Tesla P80 GPU	Camera	Not specified	Not specified	Not specified		MobileNetV3 model for edge devices,	Achieved 99.54% accuracy for the laboratory dataset and 77.71% for the realistic dataset using MobileNet V3	Accuracy drops after quantization, with a significant decrease when using float-16-bit and int-8-bit formats. Full int-8-bit quantization reduces accuracy by 0.41%.	May struggle with real-world data compared to controlled environments.