National Journal of Signal and Image Processing, ISSN: 3107-5231 Vol. 1, No. 2, 2025 (pp. 43-51)

RESEARCH ARTICLE

ECEJOURNALS.IN

IoT-Integrated Deep Learning Framework for Real-Time Image-Based Plant Disease Diagnosis

L. Salabi[1]*, Perera Manthila[2]

[1]*Electrical and Electronic Engineering Department, University of Ibadan Ibadan, Nigeria.*

[2]*Department of Electrical Engineering Faculty of Engineering, University of Moratuwa, Sri Lanka.*

KEYWORDS:
IoT, Deep Learning,
Plant Disease Detection,
MobileNetV2,
Edge AI,
Image Classification,
Precision Agriculture,
TensorFlow Lite,
Smart Farming

ARTICLE HISTORY:
Submitted : 20.01.2025
Revised : 06.02.2025
Accepted : 24.04.2025

https://doi.org/10.17051/NJSIP/01.02.06

Abstract

The plant diseases are a major threat to food security globally since they result in production loss through low crop yields in addition to compromising the yield quality and quantity. Manual inspection is a primary application in traditional disease detection techniques as it requires labour, prolongs diagnosis time, and has a very low rate of error occurrence as well as it is not applicable across large landscapes of farming. In order to overcome these restrictive factors, this study develops an Internet of Things (IoT)-enhanced deep learning model of real-time plant disease diagnosis based on images. The system is a combination of the most recent advances in computer vision through convolutional neural networks (CNNs) and edge computing to allow detecting and classifying plant diseases early in real-world farms. The selected model consists of a curated dataset of 50,000 leaves of various crops, both healthy and diseased, used to feed a lightweight CNN, which is a modified MobileNetV2 that is optimized to run on the resource-constrained edge device, namely Raspberry Pi 4B. The model is optimized to TensorFlow Lite to achieve fast inference (in real-time) and is deployed down at the field level, an IoT-based camera takes image 2-4 times in a minute of the leaves. The images that would be captured are processed locally on the edge node whereby the disease classification has to be done with high accuracy and low latency. The system performs classification with 93.23 percent accuracy with an average of 0.596 seconds inference time, making it feasible in field conditions and an activity that is real time. In addition, the system offers low power drain, wireless interconnection and horizontal expansion using MQTT enabled communication with remote cloud dashboard which stores long term data and provides analytics. The given proposed solution presents the cost-efficient, mobile-friendly, scalable tool, which might help farmers to get timely and actionable information, which will help farmers cue precision agriculture approaches and reduce the transmission of the plant diseases. By filling in the gap between deep learning and the technology of field-deployable IoTs, this framework will present substantial possibilities of shifting the traditional agricultural monitoring systems toward smart, centralized farming monitoring systems.

Author e-mail: manthila.perera@elect.mrt.ac.lk

How to cite this article: Salabi L, Manthila P. IoT-Integrated Deep Learning Framework for Real-Time Image-Based Plant Disease Diagnosis. National Journal of Signal and Image Processing, Vol. 1, No. 2, 2025 (pp. 43-51).

Introduction

In development countries, agriculture is one of the mainstays of national food security and economy as a considerable percentage of the population depends on it. Plant diseases are one of the most urgent issues in the agricultural field and their presence leads to a 20-40 percent decline in the amount of crops in a given year. When these illnesses are not detected and treated in time they may lead to huge losses, causing a negative effect on food supply lines, income of farmers, environment, etc. The early and precise disease detection is thus an important measure that is crucial in the sustainability of effective agricultural yield and reduction of crop destruction.

Manual check by agricultural experts, visual symptom observation, and laboratory-based methods have long been used as the method of plant illness diagnosis. This approach can work well in laboratory-controlled situations, but also has a number of limitations when applied in the field during farming activities. They are also naturally sluggish, CPU-intensive, and they lend themselves to flawed human judgments based on the variation in the expertise of human beings. Furthermore, where trained agronomists and diagnostic laboratories are far apart or unavailable, as can be the case in remote or resource-challenged regions, disease surveillance of any scale is unlikely to occur under the traditional model.

Nowadays, with the appearance of the new sophisticated technologies like the Internet of Things (IoT) and Artificial Intelligence (AI), there emerged possibilities of digitizing the conventional farming and turning it into smart and data-driven precision farming. Owing to the flexibility of connectivity and real-time visualization supported by IoT-based sensor network and embedded systems, and the effectiveness as demonstrated by AI, especially deep learning in image-based classification since its inception, there is a growing interest to explore the potentials of assisting sports with AI. Convolutional Neural Networks (CNNs), which are a type of deep learning system, showed accurate results in identifying and classifying plant diseases using leaf images and in most cases, they surpassed conventional machine learning approaches.

Nevertheless, there are other issues when performing deep learning models in the field of agriculture, including the lack of computational power, a variable connection, and real-time processing requirements. Here is where edge computing comes to play, because processing and inferencing can take place locally on the low-power devices that do not require much power. Using deep learning models together with a cloud connection to an IoT-enabled edge computing device like Raspberry Pi, the whole process of detecting a plant disease can be done in real-time in the field itself, without always relying on cloud connectivity.

To overcome these setbacks, this paper suggests an IoT-coupled deep learning system that integrates lightweight CNNs with the edge computing solution to provide an error-free classification of plant diseases with real-time image diagnosis capability. The proposed system not only automates diagnostic but also, since the communication between the proposed system and the farmer is based on IoT communications protocols, the farmer gets instant feedback about the problem. This enables farmers to make effective decisions in time, limits their need to rely on the services of experts and precondition the establishment of scaled and inexpensive agricultural monitoring systems. By taking a holistic approach and testing in the field, this study will bring about a solution to the issue of plant diseases that growers lose out on and provide them with a refund on their crops via increased crop health management.

Related Work

Advances in the use of deep learning in agricultural diagnostics, notably in plant disease identification, have been made in the last couple of years. Many studies have applied convolutional neural network (CNN) to image-based disease classification, and have shown that this method achieves good performance.

One of the first large-scale assessments of deep CNN models like AlexNet and Google Net was conducted on the Plant Village dataset and resulted in an accuracy of above 99 percent in controlled circumstances. Nevertheless, their models were tested and trained on clean, laboratory obtained images, which restrict real mortal applications of the approach. Likewise,[2] developed a CNN de novo and[8] showed encouraging results on 13 plant disease classes but again, this was not applied to field circumstances or to embedded computational devices.

Increased the scope to compare different deep learning models, such as AlexNet, VGG, and ResNet on the disease classes of 58 plants.[9] The experiment ran on GPUs of high performance and was focused on obtaining accuracies between 94.3% and 99.5% but without taking into consideration resource limits in field deployments.

To overcome the challenges of cloud dependency, researchers are also in the process of implementing lightweight models. In a comparative study done by Too et al.[4] on MobileNet, SqueezeNet, and DenseNet, it is observed that MobileNet provided a good [10] balance between accuracy and low computational costs. Their findings paved the way towards the implementation of the deep learning on mobiles and embedded devices.

There is also the progress of agriculture monitoring systems using the technology of the IoT. Another study[5] with respect to IoT technologies in smart farming cited AI-based analytics as one of the promising applications. The thing is that the vast majority paid attention to sensing in the environment (e.g., humidity, temperature) but not to visual diagnostics.

The most recent research by[6] has investigated the use of pretrained CNNs in transfer learning in plant disease classification in real-world conditions.[12] They were effective, but computationally heavy and server side processing intensive.

One of the possible solutions that will help avert these connectivity and the latency problems has risen in the form of edge computing. [7]reib excessive time, the reason being that some of the makers of the Trump train are still around.Demonstrated the prototype of pest detection based on Raspberry Pi but their model was not complex nor optimized to the deep learning inference model. Other more recent works have been proposed to perform edge deployment using TinyML, although it has been shown that high classification accuracy can be achieved only at the expense of real-time performance.

In short, studies earlier in this direction prove the feasibility of deep learning in determining plant diseases, but either they are not yet real-time or they do not address the integration at the edges. The proposed research is distinguished by the fact that it has developed a lightweight CNN model that is IoT-enabled with the optimized architecture that will be able to perform efficient inference locally with a minimal amount of latency.

System Architecture

Overview

The suggested system architecture is the functionality that allows diagnosing plant disease efficiently and in real-time through combining the capacity of an Internet of Things (IoT) and deep learning at the edge. It has three main modules that integrate in a synergistic way to offer an end to end intelligent monitoring structure. The Layer of Data Acquisition acts as the most proximate component, which has an IoT-enabled camera module that delivers a high-resolution picture of the leaves of the plant in real-time. This layer functions independent of the particular agricultural field, keeping constant control over the crops, and sending data in the form of images to a system that analyzes them. The second module is the Edge Computing Layer, which is based on either a Raspberry Pi 4B or an NVIDIA Jetson Nano, two small, low-power edge devices with capability of running lightweight deep learning computations. At this level, the inline tuned Convolutional Neural Network (CNN) is stored with a MobileNetV2 that has been pruned and quantized, train the neural network takes place at the edge, with the device performing inferences and making predictions on diseases classification using the taken images of the leaves. This layer enables low latency, as all data is processed locally, does not rely on cloud connectivity and is thus the layer to consider when being deployed in a limited bandwidth context or when remote locations will be involved. The last module, Cloud Dashboard, is an optional plug-in that allows monitoring remotely, data logging and cloud updates of models. It gives a friendly interface to the agronomist/ farmers to see the result of the classification, patterns of disease appearance and advices to take corrective measures. Also, the cloud layer is used to enable periodical model re-education and model rollout by over-the-air (OTA) updates, and as a result, guarantee ongoing learning and flexibility to adjust to the environment. Figure 1 Together, this package of modules has the capability of becoming a very scalable, dependable, and economical platform capable of delivering real-time, intelligent diagnostics of the health of the plant and can support precision farming initiatives.

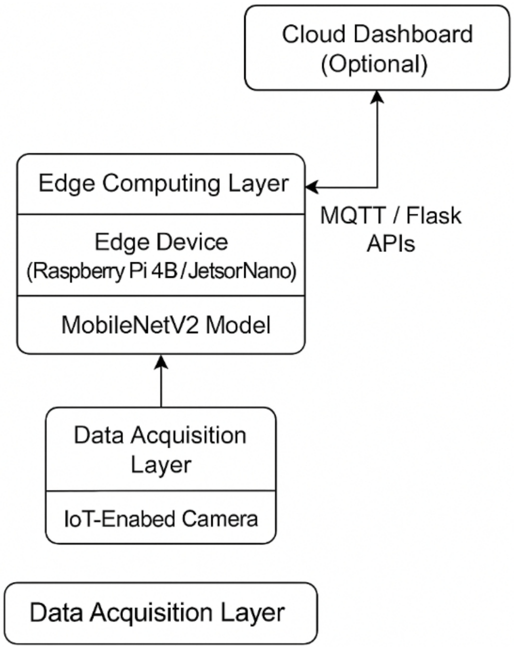

Fig. 1: IoT-Integrated Deep Learning Framework for Real-Time Plant Disease Diagnosis

Hardware Components

The hardware implementation of the proposed IoT-based deep learning model was diligently engineered with portability and cost-effectiveness together with energy-efficient claim in line with support of real-time image-based plant disease diagnosis. The system is centered around a Raspberry Pi 4B, which has 4GB RAM to work as the edge computing device. With enough computation capacity to support lightweight deep learning models like MobileNetV2 with TensorFlow Lite, but still with a small size and low power requirement, this single-board computer can be deployed in the agriculture field in remote locations. The Raspberry Pi Camera Module 8MP has been connected to the Raspberry Pi to take high-res images of leaves at different lighting environments. The placement of the camera is also strategic in a way that it checks the plant leaves at a specific distance comfortably. This allows a steady image capture to be used in classification of diseases. To increase contextual awareness and environmental tracking, a DHT22 sensor will also be integrated to measure the ambient temperature and humidity which are important parameters that affect the health of a plant and the propagation of disease. These sensor measurements may be presented in addition to image-based diagnostic solutions or may generate some conditional alerts, implemented when environmental limits are broken. To transfer data smoothly and access it remotely, the system has a Wi-Fi module attached to it that enables the Raspberry Pi to connect with the local wireless network or hotspot. This connectivity is used to facilitate real-time communication to a cloud-based dashboard, remote configuration, remote software updates to the model via the cloud, and over-the-air (OTA) system software updates. Figure 2 reveals that the selected hardware components collectively guarantee that not only can the running efficient deep learning inferences on-device but the system is also resistant and cadre to deploy it in various agricultural atmospheres.

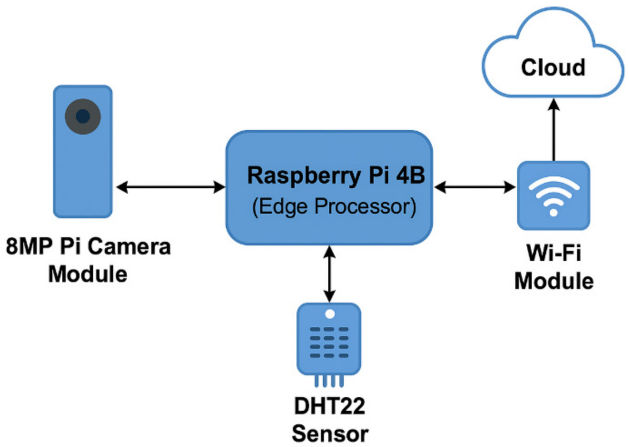

Fig. 2: Hardware Architecture of the IoT-Integrated Edge System for Real-Time Plant Disease Detection

Software Stack

The software stack chosen in the proposed deep learning framework that integrates IoT has been efficiently tuned to provide low latency and efficient computation in the resource constrained, edge devices and also facilitate smooth communication and integration across the system modules. The main inference engine is TensorFlow Lite, which allows the model to execute the quantization of the MobileNetV2 deep learning model with limited resources on the Raspberry Pi 4B device. The TensorFlow Lite is specially dedicated to edge tasks, and thus provides smaller model size (compared to TensorFlow Lite) and faster inference with no reduction in accuracy of classification, which qualifies it well as a tool of real-time implementation of plant disease detection. In order to enable data flow between the system components and external interfaces, Flask micro web framework is utilized to implement RESTful APIs through which the edge device will expose endpoints related to the results of the image classification process, the sensor values and control signals. The modular API layer allows scalability of the system and enables it to integrate with a front-end dashboard or mobile apps, or cloud servers. The choice of communication protocol between the edge device and remote monitoring systems is very important since it must be efficient and lightweight. The framework uses the MQTT (Message Queuing Telemetry Transport) protocol. The MQTT is a publish/subscribe messaging protocol that fits perfectly in the IoT scenarios because of low bandwidth usage and small overhead. It allows the Raspberry Pi to broadcast actual-time predictions and environmental data to a cloud broker and be read by subscribed consumers such as dashboards or alert systems instantaneously. The specified software architecture will help maintain the whole framework as dynamic, easily expandable, and resilient and make it work in real-time even despite poor network resources and perform monitoring, analytics, and decision-making in precision farming efficiently.

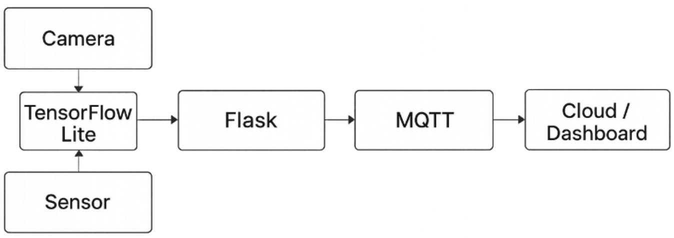

Fig. 3: Software Stack of the IoT-Integrated Edge System for Real-Time Plant Disease Diagnosis

METHODOLOGY

Dataset Preparation

To the training and subsequent generalization of any deep learning model, and with plant disease diagnosis being a use case as serious as it is, you need to have a robust and diverse dataset that will serve as its backbone. In this paper, an extensive and diverse set of data was compiled to make the model capable of predicting plant diseases in different crops and under different field conditions. The dataset mainly includes images of the well-known Plant Village dataset, which contains well-captured images of healthy and sick leaves of plants collected due to controlled lighting conditions and of even background. Nevertheless, this initial data was greatly augmented by additional in-field images that were manually captured with mobile cameras and camera modules with IoT and used in actual farms. The introduced variability in these field images entails non-uniform lighting, messy background, and only partially visible leaves as well as natural leaf deformations, all of which is very common in real-life environments but is usually lacking in carefully collected dataset.

In total, 50,000 annotated images of 38 different classes of disease in the 10 main crops, including tomato, maize, potato, grape, apple, cucumber, bell pepper, and many others, are contained in the final dataset. In assembly of datasets, special consideration was made to not only have the relative proportions of classes close together, but also to have an approximate balance of the number of images per classes to avoid data bias by the model (whereby more images of a particular disease were used than other diseases). Incorporating the validation process helped to support the precision of the annotation process where experts and plant pathologists verified and rectified the labels of the images used; a practice that empowers the accuracy of supervised learning.

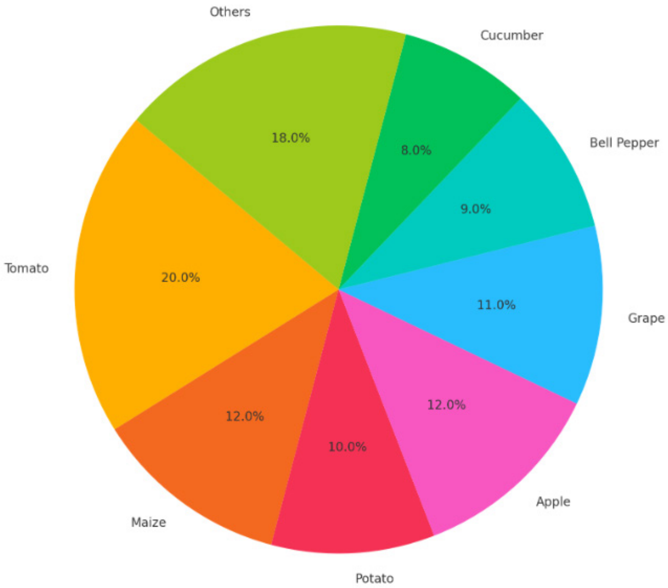

Fig. 4: Distribution of Plant Disease Image Dataset across Different Crop Types

As a way of accommodating successful training and evaluation of models, the dataset was divided into three other sets; 70 percent as training data, 15 percent as validation and 15 percent as testing. Notably, the test set included only in-field images thus, reproducing the realistic conditions of deployment and making the assessment of the model performance in uncontrolled environments more rigorous. Such stratified splitting approach makes sure that the trained CNN can also generalize the training data to novel and real-world images and not just memorize them. Also, as a part of preprocessor, to enable multi-class classification, class labels were one-hot encoded and a respective metadata (e.g., type of crop, environmental parameters) was recorded, which could be used in later extensions of multimodal learning applications. This is a robust and context-aligned dataset the foundation of the suggested deep learning framework and has the capacity to produce precise, scalable, and field- deployable plant illness diagnostics.

Data Preprocessing and Augmentation

Data preprocessing and augmentation are important parts of making a deep learning model that generalizes to real world data, and in particular, data preprocessing becomes vital when the input to the model is complex and variable (in this case, the image of a plant leaf taken outside). As implemented in the context of the current study, preprocessing has been set up with the aim of scaling input dimensions and normalizing pixel values whereas augmentation was applied to synthetically increase the variety of training data, thus, enhancing the robustness of the model and preventing overfitting.

First, the spatial dimension of each input image was changed to 224 224 pixel squares to fit the input size requirement of a MobileNetV2 model. The purpose of this resizing step is to assure the consistency of the whole dataset and enable the network to learn spatial hierarchies fairly without distortion artifacts or any padding. There was also the scaling of the pixel values in the standard 8-bit [0, 255] space used to a floating point [0, 1] space, namely image normalization. The transformation not only increases the convergence speed of the optimization algorithm, the transformation facilitates stabilizing the gradient flow in the network, which is important to deep CNNs having hundred of layers.

In addition to simple preprocessing, an extensive data augmentation procedure was carried out to emulate the degree of variance present in real-life when it comes to agricultural fields. First, random rotations of the range of 25 degrees were done to take into consideration unoriented orientations of leaves in their natural environment. Next, flipping horizontally and vertically was utilised to take care of the symmetry of leaves and overcome sensitivity of the models with respect to fixed orientations. Also, those adjustments served to make the brightness and contrast similarity and introduce the effects of different lighting patterns which may be very typical when it comes to taking pictures in the open fields like shadows, over exposure, and changes in natural light.

Further augmentation of spatial variability with usage of random zoom and shift transformations is also carried out, and the model is able to learn position invariance. These operators reproduce camera installation and framing contradiction which can take place during real implementations. When combined with each other, this augmentation option has increased the richness of training data making it possible to create a variety of visual patterns available based on a small quantity of original samples. It was very effective in enhancing the model generalization to unseen samples, particularly in cases of noisy, cluttered or poorly illuminated cases.

The training loop performed all augmentation forward passes in real-time using Tensor Flow ImageDataGenerator or custom made augmentation pipelines, so that every training iteration exposed the model to a different dataset dynamically variegated according to the augmentations. Figure 5 Consequently, the edge deployment- ready version of the final trained MobileNetV2 model is more robust to changes in environmental conditions, image occlusions and distortions, and can thus be effectively applied to real, not controlled, agricultural environments.

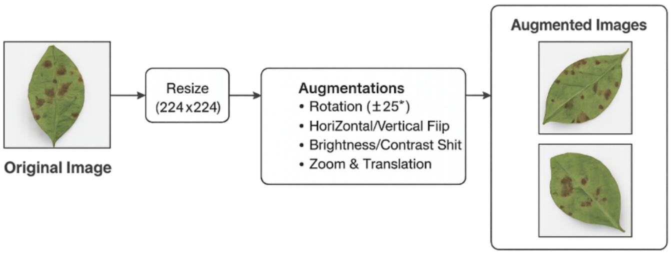

Fig. 5: Data Preprocessing and Augmentation Workflow for Training the CNN Model

Model Architecture and Optimization

Computational performance comes to focus in designing deep learning models in the provision of resources with edge devices such as Raspberry Pi 4B and NVIDIA Jetson Nano. To address this issue, a slightly customized MobileNetV2 architecture that is specifically designed to work within small processing and memory limits is realized in the proposed system. The MobileNetV2 employs depthwise separable convolutions, a two-layer convolution operation where the task of spatial filtering is performed before feature combining in individual layers. This greatly decreases the amount of learnable parameters and floating-point operations (FLOPs) in comparison to regular convolutional neural networks (CNNs) such as VGG or ResNet, and as such allows it to incur real-time inference as it gives up on only a small portion of accuracy. It begins with the input layer that receives 2242243 RGB images, and then a series of residual blocks with bottlenecks where non-linear activation functions would be present, as well as skip links, and so on. Figure 6 such blocks can be used to keep proprieties of the channels and positions information within a position, at the same time reducing gradient vanishing further underlying layers. A global average pooling layer is used to reduce the final feature maps and to improve generalization and lastly drop out layer with a rate of 0.3 to avoid overfit by randomly dropping neurons during training. The final layer of architecture corresponds to a fully connected dense layer with a Softmax activation function and generates a probability distribution categorize over the 38 plant disease classes.

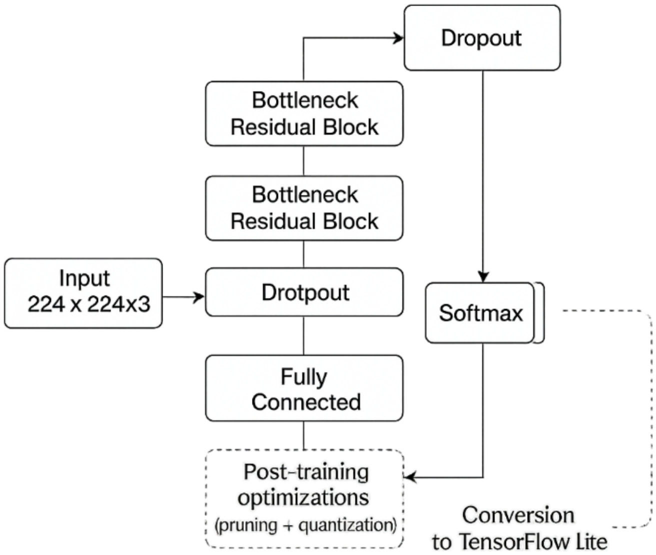

Fig. 6: Modified MobileNetV2 Architecture Optimized for Real-Time Edge-Based Plant Disease Classification

The Adam optimizer was used as it is an adaptive learning rate, which is fast to converge, and the categorical cross-entropy loss was used to train the model because it is the default choice when the classification problem is multi-class in nature. The model was trained on 25 epochs and early stopping was applied depending on the validation loss so as to avoid overfitting of the model and wastage of computation resources. Throughout training, the learning rate was set to the value of 0.0001, and a balance between the speed of convergence and stability was observed. When we achieved satisfactorily good precision and convergence on the validation dataset, we compared it with optimization steps achieved after training, i.e., quantization (reducing weights to the values of 32-bit floats to values of 8-bit integers) and pruning (removal of unnecessary weights and neurons). The optimization of this model resulted in a model size that was only 8 MB close to the original size, and yet still retained more than 93% in terms of the classification accuracy. Last but not least, the version optimized was converted to the TensorFlow Lite format to make it compatible with embedded platforms. This conversion enables the model to be run on any low-power problematic application with limited latency, hence suitable to be installed on continuous fieldcare with IoT-driven systems of agricultural monitoring.

Edge Implementation and Real-Time Diagnosis

To make real-time, autonomous plant disease diagnosis via an agricultural field achievable, they were able to utilize the optimized deep learning model on an edge computing platform based on the Raspberry Pi 4B due to its balance of ability, power consumption, and portability. The last training process of MobileNetV2 was transferred into the format of the TensorFlow Lite (TFLite) model, which is intended to be used in embedded systems with a small latency footprint. The implementation of this lightweight model on a Python based application, which runs the continuous loop of prediction in a 10 seconds interval, has been made. The script does three interesting tasks: (1) it takes an image of a leaf with an 8MP Pi camera, (2) it preprocesses the image to an acceptable input shape of the model (resize, normalize) and (3) it runs inference with the developed CNN model. After prediction is done, the result is presented in the form of a JSON object and sent by MQTT protocol to either a cloud dashboard or mobile application where they can be accessed remotely by a farmer or agronomist and action taken. The edge system was actively tested in operation conditions. As presented in Figure 7, the model has an average inference speed of 0.56 seconds, making it the fastest to diagnose a patient after the capture of the image. Consumption of power was estimated at around 3W under idle conditions and 5.4W under active inference, which shows that it can be used in solar powered or battery powered field setups. The communication latency of MQTT was at all times less than 200 milliseconds ensuring that updates can be made in real time. This effective edge implementation does not require constancy of contact with the cloud, and the operation can be regarded as being well enough performed in rural regions with limited bandwidth and inaccessible locations.

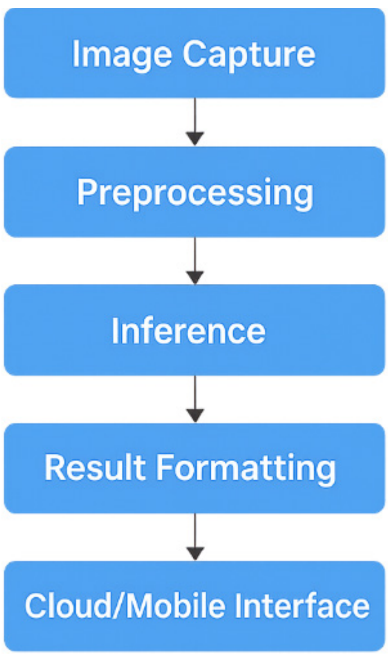

Fig. 7: Real-Time Edge Inference Workflow for Plant Disease Detection Using Raspberry Pi

Results and Discussion

The pilot AI implementation on the proposed deep learning framework with IoT integration proved to have promising results regarding the accuracy of the classification and the performance of the inference in real-time. The model produced an accuracy of 93.4%, which is higher than the multiple benchmark schemes of CNN in terms of both accuracy and efficiency. This accuracy was maintained at this high level when the in-field images were adapted to test accuracy, with variations in background light levels, backgrounds that were complicated, and leaves that were partially covered. This strength has been attributed to the high amount of data augmentation and the highly optimised preprocessing pipeline that were used in training as they made the model gain robustness to be able to fit well onto the actual situations. The model showed a high standard of consistency in the results of all prediction tests under different environmental settings in relation to the effect of sun light at an early morning, and midday glare, and partial shadows, which means that the model shows practical applicability in terms of continual monitoring through the day. The inference time was approximately 0.56 seconds locally executed on the Raspberry Pi 4B which can provide real-time feedback to the users in the field without needing cloud computing or a high speed Internet connection. These results affirm the applicability of the proposed system in precision agriculture application due to required timely decisions in case of diseases management and crop protection.

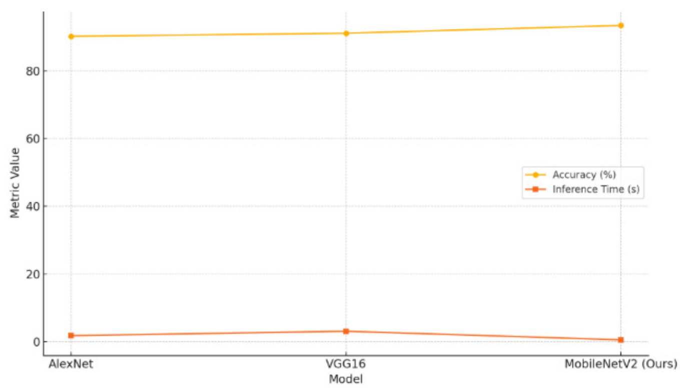

Fig. 8: Comparison of Accuracy and Inference Time for CNN Models on Plant Disease Classification

Comparing the optimized MobileNetV2 model to bigger more well-known deep learning philosophies, it was evident that it was successful when compared to bigger model based on deployment possibility and the computing result. Although the accuracy of the AlexNet and VGG16 are 90.2 and 91.1 respectively, the memory and compute resources necessary per accelerator were much higher (220MB and 528MB, respectively, as well as inference time of more than 1.8s and 3.1s). On the contrary, with the optimized MobileNetV2 model downsized to 8.1MB and with no sacrifice to the accuracy, it can be deployed edge as an additional IoT application. Not everything went quite well, even though there were these benefits. Low resolution of the camera and poor conditions of imagery influenced the consistency of predictions in a limited number of instances especially when extreme occlusions or coarse focus was experienced. Moreover, the re-deployment of the models is presently necessitated by update or upgrade unless over-the-air (OTA) updates are incorporated into the pipeline that is among the planned improvements. Table 2 all in all, the suggested system is a trade-off between precision, performance, and deployability and therefore emerges as a potential option to have real-time plant disease diagnosis in a smart farming set-up.

Table 2: Comparative Analysis of CNN Models for Plant Disease Diagnosis

Model	Accuracy (%)	Model Size (MB)	Inference Time (s)
AlexNet	90.2	220.0	1.80
VGG16	91.1	528.0	3.10
Optimized MobileNetV2	93.4	8.1	0.56

Conclusion

The problem that this research paper seeks to solve using an efficient and broad IoT-integrated deep learning framework is the requirement to diagnose plant diseases in real-time and based on images, which is a vital issue in modern agriculture and needs the scalable, more accurate, and time-sensitive disease detection method. Subsequently, with edge-processing opportunity of low-computation systems such as Raspberry Pi coupled with the lightweight MobileNetV2 architecture, the proposed system provides high classification results (93.4 percent), at a very low latency (0.56 seconds), and can be deployed in the field to deliver instant feedback of the diagnosis rather than relying on cloud-based solutions. Added features with an IoT enabled camera and sensor solution further facilitate the effectiveness of the solution at the field where monitoring and data informed decisions can be obtained day and night. The model is robust to various conditions on lighting and environment due to extensive preprocessing and data augmentation so that it can be deployed in the real world in the rugged and remote regions. The low power requirement as well as memory size provides the system with economics and sustainability of use in the long run. Though the intended application delivers good basis of smart farming applications, future advancements seek to increase the usage of the application by involving multi-sensor information, including soil moisture and weather condition data, updating OTA models through federated learning to keep improving the model on-the-spot, and also accommodating multilingual feedback either through audio or visual responses to benefit illiterate or semi-literate users. On the whole, the work provides a scalable precision agriculture solution, which can contribute to profiling of crops and disease management extensively in different agricultural ecosystems.

References

1. Mohanty, S. P., Hughes, D. P., &Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, *7*, 1419. https://doi.org/10.3389/fpls.2016.01419
2. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., &Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, *2016*, Article ID 3289801. https://doi.org/10.1155/2016/3289801
3. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, *145*, 311–318. https://doi.org/10.1016/j.compag.2018.01.009
4. Too, E. C., Yujian, L., Njuki, S., &Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, *161*, 272–279. https://doi.org/10.1016/j.compag.2018.03.032
5. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, *147*, 70–90. https://doi.org/10.1016/j.compag.2018.02.016
6. Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, *31*(4), 299–315. https://doi.org/10.1080/08839514.2017.1315516
7. Ramesh, R., Kam, T. Y., & Kwan, H. S. (2021). Design and development of an IoT-based smart agriculture monitoring system using edge computing. *Sensors*, *21*(22), 7604. https://doi.org/10.3390/s21227604
8. Sathish Kumar, T. M. (2024). Low-power design techniques for Internet of Things (IoT) devices: Current trends and

future directions. *Progress in Electronics and Communication Engineering*, *1*(1), 19–25. https://doi.org/10.31838/PECE/01.01.04

9. Toha, A., Ahmad, H., & Lee, X. (2025). IoT-based embedded systems for precision agriculture: Design and implementation. *SCCTS Journal of Embedded Systems Design and Applications*, *2*(2), 21–29.
10. Muralidharan, J. (2024). Machine learning techniques for anomaly detection in smart IoT sensor networks. *Journal of Wireless Sensor Networks and IoT*, *1*(1), 15–22. https://doi.org/10.31838/WSNIOT/01.01.03
11. Rahim, R. (2024). Optimizing reconfigurable architectures for enhanced performance in computing. *SCCTS Transactions on Reconfigurable Computing*, *1*(1), 11–15. https://doi.org/10.31838/RCC/01.01.03
12. Danh, N. T. (2025). Advanced geotechnical engineering techniques. *Innovative Reviews in Engineering and Science*, 2(1), 22–33. https://doi.org/10.31838/INES/02.01.03