

Automatic Early Detection of Tomato Leaf Disease using IoT and Deep Learning

I. Sheik Arafat

sheik.arafat@gmail.com

Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology

S. Aswath

Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology

S. M Haji Nishath

Mohamed Sathak Engineering College

Research Article

Keywords: Internet Of Things, Deep Learning, Mobile Application

Posted Date: March 21st, 2024

DOI: <https://doi.org/10.21203/rs.3.rs-3958296/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

Tomato plants are defenseless to different illnesses, including bacterial, contagious, and viral contaminations, which can fundamentally lessen crop yield and quality on the off chance that not identified and treated early. Farmers may experience increased crop damage and financial losses as a result of this detection delay. The goal is to foster a robotized framework utilizing IoT (Internet of Things) gadgets, for example, cameras conveyed in the field, joined with profound learning strategies, to precisely and quickly distinguish illnesses in tomato plants. This framework intends to give ranchers an early admonition framework that can recognize and order infections quickly, empowering convenient intercession and designated treatment, accordingly further developing harvest wellbeing and yield. Profound learning has essentially expanded the precision of picture classification and article identification frameworks' acknowledgment as of late. The exploration zeroed in on computerizing the early location of tomato leaf sicknesses utilizing IoT innovation and a changed ResNet50 profound learning model. At first, IoT gadgets, including sensors and cameras, were conveyed in tomato fields to gather plant-related information and pictures. We focused on calibrating the hyper boundaries of pre-prepared models, including GoogLeNet, SqueezeNet and ResNet-50. The notable Tomato leaf disease detection dataset, which incorporates 3,890 picture tests of different sickness and healthy leaves, was utilized for the tests. Using comparable cutting-edge research, a comparative analysis was also conducted. The tests showed that ResNet-50 outflanked cutting edge models with a 99.87% more prominent characterization exactness. The framework demonstrated commendable capability in identifying whether tomato plant leaves were affected by disease in their early stages. This capability enabled farmers to receive timely alerts through mobile application, allowing for more effective management of the issue.

1. Introduction

Tomato plants hold significant agricultural importance globally, including in India, owing to their versatile culinary uses and nutritional value. In India, tomatoes are a staple in various regional cuisines and a vital ingredient in many dishes. Additionally, they contribute significantly to the country's economy as a high-demand cash crop. Tomatoes are rich in vitamins, antioxidants, and minerals, making them essential for a balanced diet. Moreover, their adaptability to diverse climatic conditions makes them widely cultivated across the world, ensuring food security and income for farmers. The global demand for tomatoes continues to rise, not only for fresh consumption but also for processed products like sauces and canned goods. As such, the cultivation and health of tomato plants are paramount in India and worldwide agriculture, impacting food availability, nutritional intake, and economic stability.

Tomatoes are the most popular canned vegetable, despite being the third-largest vegetable crop in the world after sweet potatoes and potatoes. Total acreage planted to tomatoes worldwide is 46.16 lakh hectares, and the crop yield is 1279.93 lakh tons. The area, productivity, production, and percentage share of the world's total production of tomatoes among the main producing nations.

Diseases regularly crop up despite proper management measures, which is one of the biggest problems for organic tomato growers. The degree of incidence is mostly determined by environmental factors and is depending on regional factors. Pathogenic infections produced by bacteria and fungus as well as physiological problems like blossom end rot and cat facing, which are brought on by environmental or abiotic stress, can harm tomatoes. Pathogenic illnesses can arise from infections both above and below the ground, along with insect feeding. Major afflictions in tomatoes encompass those affecting above-ground components such as stems and leaves (e.g., bacterial canker, verticillium wilt, late blight, rhizoctonia, bacterial wilt), the fruit (e.g., bacterial spot, bacterial speck, anthracnose), and the root system (e.g., fusarium wilt, bacterial wilt, rhizoctonia). Therefore, implementing a comprehensive disease control program is essential at every developmental stage. Utilizing resistant cultivars can effectively counter bacterial

diseases, and implementing regular crop rotation with non-solanaceous plants aids in preventing such ailments. Consequently, it is not recommended to cultivate tomatoes, potatoes, chilies, and brinjals consecutively.

Detecting diseases at their onset allows for timely intervention and management strategies, leading to several critical advantages:

1. Early detection enables prompt action, facilitating preventive measures such as targeted pesticide application, crop rotation, or use of resistant varieties. This helps in containing the spread of diseases and limiting crop damage.
2. Prompt identification and intervention mitigate the impact of diseases, preventing extensive damage to the crops. Minimizing the spread of infections at an early stage contributes to maintaining higher yields and overall productivity.
3. Early detection strategies are more cost-effective than attempting to control established diseases. Timely management reduces the need for excessive pesticide use or expensive treatments, saving farmers both money and resources.
4. Diseases often affect the quality of crops, making them less marketable or suitable for consumption. Early detection and management help maintain the quality and market value of the produce.

IoT and deep learning technologies have revolutionized agriculture, offering innovative solutions to enhance productivity, sustainability, and efficiency in farming practices. The integration of IoT-generated data with deep learning algorithms offers smart farming solutions. This combination enables predictive models for disease detection, yield estimation, and irrigation management, facilitating more efficient and sustainable agricultural practices. IoT devices collect real-time data, which is processed by deep learning models for immediate analysis. This synergy allows for timely responses, such as automated alerts for disease outbreaks or optimized resource allocation based on current field conditions.

The integration of IoT and deep learning technologies presents significant opportunities for the modernization of agriculture. This convergence facilitates data-driven decision-making, optimizes resource utilization, and contributes to enhanced productivity and sustainability in farming practices. Significant progress in the field of agriculture has been made in recent times, primarily due to the general enthusiasm surrounding deep learning (DL) and the Internet of Things (IoT) (Gracia et al., 2021). The Internet of Things is essential for gathering up-to-date data and encouraging wise use of electricity, water, and fertilizer (Kundu et al., 2021). In addition, DL methods work well for tasks including recognising objects, classification, matching patterns, and picture recognition (Dhaka et al., 2021). In areas like finding plant illnesses (Mohanty et al., 2016), weed recognition (Fawakherji et al., 2019), fruit counting (Chen et al., 2017), estimation of yield (Nevavuori et al., 2019), and visualisation of detected fruits, diseases, or weeds (Chen et al., 2017). Deep Convolutional Neural Networks (DCNNs) demonstrate their efficacy in the automated extraction and selection of features. By using these methods, image recognition and feature extraction can be done more quickly and precisely (Kamilaris et al., 2017).

This paper is structured into the following sections. A review of the literature on illness detection using DL and IoT is presented in Section 2. The latest developments for several kinds of deep learning models are described in Section 3. The study and application of IoT and three different DL models for tomato leaf disease diagnosis are the primary focal points of Section 4. In Section 5, the conclusion and directions for further study are covered.

2. Literature Survey

Ignoring early warning indicators of plant disease in farming can lead to lower food yields, which in turn has a negative impact on the world economy (Upadhyay et al., 2022). An extensive summary of recent advances in research on plant leaves is given in this section.

A paper, referred to as (Upadhyay et al., 2022), introduced a deep learning model based on CNN, which aimed to achieve precise classification of plant diseases. An enormous dataset consisting of 87,000 RGB photos that were made freely available was used to train the model. First, there was preprocessing, and then there was segmentation. Despite achieving a 93.5% identification accuracy rate, the CNN had trouble classifying certain groups, which caused confusion later on. Furthermore, the model's performance decreased as a result of low data availability. To enhance the accuracy of recognition, (Eunice et al., 2022) proposed a hybrid CNN-SVM technique for the classification of diseases in banana plants. In their study, they applied a median filter to preprocess the raw input image, while keeping the standard image dimensions and default settings intact. This hybrid approach combined an SVM with a CNN. The health condition of banana leaves was first identified using a single SVM in phase 1, and then a multiclass SVM was employed in the testing phase to detect specific diseases or infections within the affected leaves. By feeding the output of the CNN into the support vector machine in this manner, a classification accuracy of 99% was achieved. Previous studies have demonstrated that CNNs are more accurate than traditional techniques, although they have been limited in terms of their diversity of approaches.

(Jadhav et al., 2021) proposed the use of a CNN for the detection of plant diseases, leveraging pre-trained models that have been trained on identifying illnesses in soybean plants. While transfer learning methods like AlexNet and GoogLeNet yielded superior results, the model struggled to diversify classifications. Due to resource limitations for training various deep learning models, current models tend to focus on recognizing specific classes of plant diseases rather than developing a comprehensive model for classifying numerous plant disorders.

In their study, (Abayomi-Alli et al., 2021) introduced a novel histogram modification method to generate synthetic image samples from low-quality test set photos, thereby enhancing the recognition performance of deep learning models. To overcome data scarcity during training and improve photo quality, a modified MobileNetV2 neural network model was utilized to edit photographs in the cassava leaf disease dataset. Various techniques such as Gaussian blurring, motion blurring, resolution down-sampling, and over-exposure were employed.

(Abbas et al., 2021) simplified the previously expensive, labor-intensive, and challenging process of real-time data collection and acquisition by developing a conditional generative adversarial network that generates simulated images of tomato plant leaves. (Anh et al., 2021) demonstrated a classification model based on a pre-trained Mobile Net CNN model using a benchmark dataset, achieving a reliable accuracy of 96.58%. Furthermore, a multi-label CNN using transfer learning methods including DenseNet, Inception, Xception, ResNet, VGG, and MobileNet was proposed in (Kabir et al., 2022) for the classification of different plant diseases.

Another suggestion in (Astani et al., 2022) was to evaluate the efficacy of the Ensemble Classifier in classifying plant diseases using datasets such as Plant Village and Taiwan Tomato Leaves. Resilient convolutional neural network in (Gokulnath et al., 2021) failed to perform well with real-time photographs collected in different environmental conditions using Tomato leaf disease detection benchmark dataset, indicating limitations in its applicability.

3. Methodology

In the realm of agricultural disease detection, the utilization of Internet of Things (IoT) devices, particularly cameras, holds significant importance in the collection of visual data for assessing plant health. The following elucidates the application of IoT cameras in agricultural data collection:

Cameras are strategically positioned on fixed mounts at specified locations within the field to facilitate data gathering. The placement of these cameras is contingent upon factors such as field size, crop type, and the targeted monitoring area. To cover expansive regions or specific points of interest within the crop field, multiple cameras may be deployed.

RGB cameras capture images of tomato plants within the visible spectrum, offering crucial color information for visual inspection and disease identification based on color variations, lesions, or anomalies on plant leaves. The cameras can be configured for continuous monitoring, capturing images at predetermined intervals (e.g., daily). Subsequently, the acquired images and relevant data are wirelessly transmitted to a cloud server through a LoRa gateway equipped with a Raspberry Pi. Upon transmission to the cloud, tomato plant images undergo a sophisticated segmentation process via Mask R-CNN, a technique prominently featured in the work of (Yang et al., 2020). This model is chosen for its exceptional precision in separating objects from complex backgrounds, particularly beneficial for scenes involving tomato plants against challenging backgrounds.

Mask R-CNN is employed for the purpose of segmenting the leaf image from the overall plant image. This deep learning segmentation algorithm, known as Mask R-CNN, finds its utility in a wide range of applications, particularly in instance segmentation. It is an extension of Faster R-CNN and excels in both object detection and pixel-level segmentation, making it well-suited for tasks involving the dissection of overlapping plant leaves. The initial step in this process entails the utilization of a pre-trained "backbone network" such as VGG16 or ResNet, which extracts feature maps of high quality from the entire image. Subsequently, the "Region Proposal Network (RPN)" acts as a scout, generating potential object locations (bounding boxes) for further investigation. Following this, the "ROI Pooling" stage isolates regions of fixed size from the feature maps, thereby directing the analysis towards each proposal. This prepares the proposals for the final step, wherein the "Mask Branch" takes center stage and employs a separate neural network to predict a segmentation mask for each proposal. This intricate mask meticulously delineates the precise shape of each individual leaf, even in the presence of overlapping foliage. Concurrently, the "Classification Branch" identifies the object class associated with each proposal. This dual functionality provides both accurate segmentation and classification, thereby proving invaluable for the analysis of plant leaves. The strengths of Mask R-CNN lie in its ability to accurately segment individual objects, even in the presence of complex backgrounds; its efficiency in leveraging pre-trained models for feature extraction; and its versatility in handling diverse datasets containing multiple leaf species and overlapping structures. However, like any demanding concerto, Mask R-CNN has certain requirements. Large datasets are crucial for achieving optimal performance, and the training process can be computationally expensive, particularly for large datasets. Additionally, fine-tuning the hyperparameters and selecting the appropriate model are essential for achieving peak performance. Mask R-CNN serves as a powerful tool for unraveling the hidden secrets within plant images. With its precise segmentation and classification capabilities for individual leaves, it proves to be a valuable asset for various tasks related to plant leaf classification and analysis. Therefore, in this work, Mask R-CNN plays a pivotal role in the segmentation of the leaf from the plant image before subjecting it to classification. Through this approach, the image is effectively segmented into distinct regions, laying the groundwork for subsequent analyses and applications in plant classification and monitoring. The segmented tomato leaf image is further employed for disease classification.

For disease classification, pretrained deep learning models, including GoogLeNet, SqueezeNet, and ResNet-50, are utilized, and their classification efficiency is compared. The best-performing model among the three, ResNet-50, is selected for deployment on the cloud server to classify the images. The Tomato leaf disease detection dataset, comprising 3890 images (1926 healthy and 1974 diseased), is employed to train and test the models. The entire dataset was split in the experiment in an 80:20 ratio between training, testing, and validation data.

Comparative analysis highlights the remarkable superiority of ResNet-50 in terms of classification accuracy and other performance metrics. The classified information is promptly updated in the cloud, and an alert mechanism is implemented to notify farmers on their smart devices. This provides real-time insights into detected diseases, their locations, and recommended proactive measures. The selection of ResNet-50 as the deep learning model is justified by its outstanding performance, ensuring accurate and effective detection of tomato leaf diseases in agricultural settings.

Figure 1 provides direction for creating an automated solution that combines DL and IoT methods to help farmers practice smart farming.

Conventionally, the IoT architecture comprises five layers (Indira et al., 2023): a physical layer, network layer, middleware layer, processing layer, and application layer. In our proposed work, we incorporate an IoT layer architecture as

1. Device Layer:

Sensor Nodes are Deployed in the field, these sensor nodes consist of various sensors like IR cameras, Visual Cameras and other relevant sensors. Each sensor node is equipped with a Raspberry Pi board acting as an IoT gateway, capable of processing and transmitting data. LoRa Transceivers are Connected to the Raspberry Pi and it facilitate long-range wireless communication for transmitting sensor data. These transceivers transmit data wirelessly from the sensor nodes to the gateway.

2. Connectivity Layer:

A central LoRa gateway equipped with a Raspberry Pi collects data transmitted by sensor nodes using LoRa transceivers. LoRaWAN or other suitable communication protocols are used for efficient data transmission between sensor nodes and the gateway.

3. Network Layer:

Connectivity Backbone layer establishes the connection between the LoRa gateway and the cloud or central server. Utilizes suitable network protocols and infrastructure to ensure reliable data transfer from the gateway to the cloud.

4. Cloud Layer:

Cloud Server Receives and processes data transmitted by the gateway. Storage Systems which store the incoming data in databases or data lakes for further analysis and processing. Initial data processing occurs here, including data cleaning, normalization, and pre-processing for use in the Deep Learning model.

5. Application Layer:

Deep learning trained model for Leaf disease detection is deployed and operated in this layer. These model analyze the pre-processed data to detect pests. Based on the analysis from the Deep Learning model, decisions are made regarding the presence of pests or potential threats. If diseases are detected, alerts or notifications are generated for the concerned stakeholders through various means of mobile applications. Mobile application also consists of various other modules along with disease detection.

- Although the health of a crop is estimated through field monitoring and information collected by other devices, such as IR cameras, visual cameras, etc., DL algorithms can operate on such live data sets obtained by IoT

devices (Patil et al., 2021). Early detection and examination of these problems is highly beneficial for crop disease prevention. Together, IoT and DL techniques enable automated pest identification and alerts or warnings.

- Here, we offer three distinct deep learning models that are appropriate for identifying leaf disease in tomato plants.

GoogLeNet/Inception

The GoogLeNet architecture was introduced in 2014 (Team, O.T.R, 2021) as a response to the resource utilization and parallel processing challenges faced by the AlexNet model. This architecture consists of 22 layers, with 21 convolutional layers and 1 fully connected layer. It incorporates batch normalization to address the issue of vanishing gradients and has four million trainable parameters. The softmax activation function is utilized for multiclass classification, with 10 output units. GoogLeNet has proven to be effective, achieving a top-five test error rate of 6.67% in the ILSVRC-2014 competition, which is 8.63% lower than AlexNet. Notably, this model replaces fully connected layers with global average pooling, and its performance can be improved by increasing its limit of divergence.

SqueezeNet

In 2017 (Iandola et al., 2016), the SqueezeNet model was introduced with the goal of developing a CNN model that has low computational complexity while maintaining high accuracy. This model uses 1×1 filters instead of 3×3 filters and is composed of fire modules and pooling layers. Each fire module includes a squeeze layer to reduce network depth and an expanded layer to increase it. Both layers have feature maps of the same size. SqueezeNet has 1.25 million trainable parameters and is claimed to be 50 times shallower than AlexNet. The addition of a bypass connection to the network resulted in a 2.9% increase in top-one accuracy and a 2.2% increase in top-five accuracy.

ResNet

To address the challenges of training deeper CNNs, which require extensive datasets and prolonged training times, researchers introduced residual networks in 2015 (He et al., 2016). ResNet has variations of 50, 101, or 152 layers, with the number of trainable parameters varying accordingly. ResNet employs residual functions instead of traditional learning methods to address the vanishing gradient problem. The softmax layer is integrated into ResNet for plant disease identification. However, these models require extended training periods, which can be challenging for real-time applications that require quick decision-making.

ResNet-50, which consists of 50 deep layers organized into five stages comprising convolution and identity blocks, serves as a backbone for computer vision tasks. ResNet introduces the concept of stacking convolution layers with skip connections to address the vanishing gradient issue. By bypassing the original input to reach the output, ResNet mitigates gradient problems. Placing the skip connection before the activation function further helps to minimize gradient issues, acknowledging that deeper models may encounter more errors. The shortcut connections in the residual neural network are based on identity mapping, providing a solution for these challenges.

Let x represent the input image, $H(x)$ the residual mapping, and $F(x)$ the mappings that suit the nonlinear layers. The residual mapping formula can therefore be expressed as follows:

$$H(x) = F(x) + x \quad (1)$$

Convolution is an identification block in ResNet-50. Each identity block consists of three convolutional layers and about 23 million trainable parameters. The two matrices, input x and shortcut x , can only be joined when the output

dimension from a shortcut and the convolution layer—post-convolution and batch normalization—are the same. If not, shortcut x must pass via a convolution layer and batch normalisation in order to ensure dimensional alignment.

Table 1 illustrates typical configurations for each model. Variants may possess distinct structures and performance metrics. SqueezeNet employs fire modules, which merge 1x1 and 3x3 convolutional layers to achieve efficient feature extraction. GoogLeNet's inception modules facilitate parallel processing of various filter sizes, thereby enhancing flexibility. ResNet's residual blocks tackle the issue of the vanishing gradient by introducing shortcut connections, enabling the creation of deeper networks.

The selection of a model is contingent upon specific requirements:

Feature	SqueezeNet	GoogLeNet	ResNet-50
Overall Architecture	Fire modules with squeeze and expand paths	Inception modules with parallel convolutional layers	Residual blocks with shortcut connections
Layers (Total)	53	22	174
Convolutional Layers	18 (within fire modules)	22 (within inception modules)	164 (within residual blocks)
Pooling Layers	4 Max Pooling	5 Max Pooling, 2 Global Average Pooling	3 Max Pooling
Batch Norm Layers	8	2, 2 Global Average Pooling	0- (implicitly through residual connections)
Fully Connected Layers	1	1	1
Other Layers	Dropout	-	Activation functions (ReLU, Sigmoid)
Learnable Parameters	~736,450	~6 million	~23.5 million
Image Input Size	227x227	224x224	224x224
Target Outputs	1000 classes (ImageNet)	1000 classes (ImageNet)	1000 classes (ImageNet)
Advantages	Very lightweight, computationally efficient	Good accuracy, innovative architecture	High accuracy, addresses vanishing gradient problem
Disadvantages	Lower accuracy than other models	Complex structure, computationally expensive	Large number of parameters, requires more training data

Accuracy: ResNet50 generally attains the highest level of accuracy. Computational Cost: Squeeze Net is notably more lightweight and exhibits faster performance. Training Data: ResNet50 necessitates a larger amount of data due to its larger parameter count. Implementation Complexity: GoogLeNet's inception modules may present more intricate challenges when implementing in real time.

Hence, we have opted for ResNet50 as it delivers satisfactory accuracy and is comparatively easier to implement in real-time scenarios.

4. Analysis and Implementation of Suitable Deep Learning Methodology:

4.1. Dataset

The sample included ten distinct illness classifications. Nine different varieties of tomato leaves were infected, and one class of leaves proved resistant. As seen in Figure 2, we identified our dataset classes using reference photographs and disease names.

The model we trained and evaluated here is for binary classification, meaning it will determine whether a given leaf is healthy or diseased. As the less weight pre-trained models don't inherit the higher accuracy and precision needed for multiclass classification, it isn't completed yet. Our aim is to detect the presence of a disease and provide assistance to the farmer by means of mobile application.

In pursuit of our objective to attain a high-performance, low-complexity model with faster classification speed, we have successfully developed a binary classification model. This model is specifically designed for seamless integration into a cloud server, requiring minimal memory resources for efficient loading.

4.2 Experiment:

Three distinct architectures with different learning rates were employed in our investigation of the detection of tomato leaf disease by means of transfer learning: ResNet 50, GoogleNet, and SqueezeNet. In order to obtain valuable insights, a comprehensive analysis was carried out on the assessment metrics, which encompassed True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), Accuracy, Precision, Recall (also known as Sensitivity / TPR), F1 Score, Specificity (TNR), False Positive Rate (FPR), and False Negative Rate (FNR).

For the purpose of further training the deep CNN networks that were previously trained, a publicly accessible dataset known as Tomato leaf disease detection (Dataset, 2020) was utilized. Each model was standardized to two distinct learning rates (0.01% and 0.001%) for our experimental setup.

The dataset was divided into training, test, and validation samples, where 80% of the Tomato leaf disease detection samples were used for training the pre-trained Inception, SqueezeNet, GoogLeNet, and ResNet 50 models. Each model underwent a 20-epoch run.

The experiment results and the Confusion Matrix for ResNet 50, employing two distinct learning rates of 0.01 and 0.001, are depicted in Figure 3.a, 3.B, and 3.C.

At a learning rate of 0.001, ResNet 50 exhibited exceptional performance, achieving an accuracy of 99.87%. The model demonstrated an exemplary balance between precision (99.75%) and recall (100%), resulting in an F1 score of 99.87%. Notably, the false positive and false negative rates were minimal (0.25% and 0%, respectively). Especially False Negative rate is the one which we are much interested in as negative represents healthy one here. This underscores the robustness of ResNet 50 in correctly classifying both healthy and diseased tomato leaves. Interestingly, increasing the learning rate to 0.01 did not adversely affect the model's performance, instead it enhanced the model's performance by maximizing all the metrics. Thus, emphasizing its stability and resilience to hyperparameter variations.

The Experiment result and Confusion Matrix of GoogLeNet for two different learning rates of 0.01 and 0.001. shown below in Figure 4.a ,4.B, 4.C

The GoogLeNet architecture demonstrated strong capabilities at a learning rate of 0.001, achieving a commendable accuracy of 99.62%. Precision, recall, and F1 score were all notably high (99.49%, 99.75%, and 99.62%, respectively), reflecting effective disease classification. The model showcased consistency when the learning rate was increased to

0.01, maintaining high accuracy and consistent precision, recall, and F1 score values. This resilience to changes in the learning rate suggests the reliability of GoogLeNet for the binary classification task.

SqueezeNet exhibited comparatively lower accuracy at 75.13% with a learning rate of 0.001. Precision (74.94%) and recall (76.46%) indicated a balanced performance, although the false positive rate (26.24%) and false negative rate (23.44%) were relatively high. Increasing the learning rate to 0.01 led to an improvement in accuracy (79.62%) but revealed sensitivity to hyperparameter adjustments, with shifts in false positive and false negative rates. These findings suggest that SqueezeNet's performance is more responsive to variations in the learning rate compared to ResNet 50 and GoogLeNet. So this model is not suitable for Leaf disease detection.

4.4 Result & Analysis

ResNet 50 and Google Net emerged as robust and high-performing models for tomato leaf disease detection. The consistent excellence of ResNet 50 across different learning rates as well as the near zero FPR and FNR makes it an attractive choice, while Google Net demonstrated reliability and resilience. In contrast, SqueezeNet, while offering competitive performance, exhibited a degree of sensitivity to the learning rate.

In conclusion, the evaluation of three prominent convolutional neural network architectures ResNet 50, GoogLeNet, and SqueezeNet revealed distinct performance characteristics in classifying healthy and diseased tomato leaves. ResNet 50 stood out with remarkable accuracy (99.87%) at a learning rate of 0.001, demonstrating a robust ability to precisely identify both healthy and diseased instances. Notably, even with an increased learning rate of 0.01, ResNet 50 maintained and, in fact, enhanced its performance metrics, emphasizing its stability and resilience to hyperparameter variations. Google Net showcased consistent high performance across various learning rates, affirming its reliability for the binary classification task. Conversely, SqueezeNet exhibited comparatively lower accuracy at 75.13% with sensitivity to learning rate adjustments, indicating its performance responsiveness to hyperparameter variations. Performance characteristics of three distinct architectures ResNet 50, GoogLeNet, and SqueezeNet are shown in below table2 and Figure 5.

Table 2. Performance Comparison of ResNet 50, GoogLeNet, and SqueezeNet in two different learning rates 0.001 & 0.01

Model Name	Learning Rate	TP	TN	FP	FN	Accuracy	Precision	Recall (Sensitivity / TPR)	F1 Score	Specificity (TNR)
ResNet 50	0.001	395	384	1	0	99.87	0.997	1	0.998	0.997
	0.01	395	385	0	0	100.00	1	1	1	1
GoogLeNet	0.001	394	383	2	1	99.62	0.995	0.997	0.996	0.994
	0.01	394	385	0	1	99.87	1	0.997	0.998	1
SqueezeNet	0.001	302	284	101	93	75.13	0.75	0.765	0.76	0.737
	0.01	312	309	76	83	79.62	0.80	0.79	0.796	0.802

Findings from above table underscore the nuanced strengths and adaptability of each architecture in the context of tomato leaf disease classification. These observations offer insightful advice that will be very helpful in choosing and optimizing deep learning models for agricultural disease detection applications. The study opens the door for more investigation and useful application by highlighting the significance of architectural selection and hyperparameter optimization.

4.5 User Interface Implementation

After the ResNet50 model successfully classifies leaf diseases, the subsequent step involves transmitting the results to a mobile app for timely notifications. Following the classification, the application formulates a notification message that encompasses crucial details such as the classification outcome (healthy or diseased), confidence levels, and potentially an image of the affected leaf. This message is then dispatched via an API request to a designated notification service, with each mobile device associated with a unique token for precise delivery. The notification service processes the API request, identifies the target device using the provided token, and delivers the notification to the respective mobile app. The app, consistently monitoring for incoming notifications, interprets the message payload and displays the notification to the user. Notification systems like Apple Push Notification systems (APNs) for iOS and Firebase Cloud Messaging (FCM) for Android are frequently used for this purpose. To engage with the notification, users can open the app and see further information, such as the number and location of the leaves. The practical implementation of this model is shown in figure 6

5. Conclusion

Implementation of a comprehensive system for the identification of leaf diseases using the ResNet50 model, LoRaWAN technology for image transfer, and mobile app notifications represents a robust and efficient approach to precision agriculture. The ResNet50 model exhibited exceptional accuracy in classifying healthy and diseased tomato leaves, showcasing its resilience to hyperparameter variations. Leveraging LoRaWAN for image transmission from field devices to the cloud ensures seamless data transfer, while the integration of a notification system with the mobile app provides real-time updates to users. This interconnected system not only enhances the monitoring and management of crop health but also demonstrates the adaptability of deep learning models in the agricultural domain. The presented framework not only facilitates prompt decision-making for farmers but also serves as a scalable and adaptable solution for broader applications in smart agriculture and crop management. As technology continues to advance, the synergy between deep learning, IoT, and mobile applications holds significant promise for revolutionizing agricultural practices, making them more informed, efficient, and sustainable.

In future research, this work can be extended in several key directions to enhance the proposed framework for crop disease identification. Firstly, exploring multi-class classification capabilities of the deep learning model would broaden its applicability by accommodating a more diverse range of diseases and pests. Additionally, investigating transfer learning and model fine-tuning techniques could optimize the model's performance for specific crops and regional nuances. Real-time monitoring and feedback features can be further developed to enable continuous crop surveillance, ensuring prompt interventions in response to disease outbreaks. Through these future endeavors, the framework has the potential to evolve into a versatile, user-friendly, and impactful tool for precision agriculture.

Declarations

Funding Declaration:

This research did not receive any funding from governmental, private, or nonprofit organizations.

Data Availability declaration:

The data utilized in this research is available from its original source, as cited in the article. The PlantVillage dataset can be accessed at the following URL: <https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf?resource=download>

Competing Interest declaration:

The authors declare no competing interests

Ethics approval and consent to participate:

Not applicable

References

1. Abayomi-Alli, O. O., Damaševičius, R., Misra, S., & Maskeliūnas, R. (2021). Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. *Expert Systems*, 38(7), e12746.
2. Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. *Computers and Electronics in Agriculture*, 187, 106279.
3. Anh, P. T., & Duc, H. T. M. (2021, October). A benchmark of deep learning models for multi-leaf diseases for edge devices. In *2021 International Conference on Advanced Technologies for Communications (ATC)* (pp. 318-323). IEEE.
4. Astani, M., Hasheminejad, M., & Vaghefi, M. (2022). A diverse ensemble classifier for tomato disease recognition. *Computers and Electronics in Agriculture*, 198, 107054.
5. Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., ... & Kumar, V. (2017). Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters*, 2(2), 781-788.
6. Dataset, Tomato leaf disease detection, 2020 <https://www.kaggle.com/datasets/kaustubhb999/tomatoleaf?resource=download>
7. Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Ijaz, M. F., & Woźniak, M. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors*, 21(14), 4749.
8. Eunice, J., Popescu, D. E., Chowdary, M. K., & Hemanth, J. (2022). Deep learning-based leaf disease detection in crops using images for agricultural applications. *Agronomy*, 12(10), 2395.
9. Fawakherji, M., Youssef, A., Bloisi, D., Pretto, A., & Nardi, D. (2019, February). Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation. In *2019 Third IEEE International Conference on Robotic Computing (IRC)* (pp. 146-152). IEEE.
10. García, L., Parra, L., Jimenez, J. M., Lloret, J., & Lorenz, P. (2020). IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture. *Sensors*, 20(4), 1042.
11. Gokulnath, B. V. (2021). Identifying and classifying plant disease using resilient LF-CNN. *Ecological Informatics*, 63, 101283.
12. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
13. Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.
14. Indira, P., Arafat, I. S., Karthikeyan, R., Selvarajan, S., & Balachandran, P. K. (2023). Fabrication and investigation of agricultural monitoring system with IoT & AI. *SN Applied Sciences*, 5(12), 322.
15. Jadhav, S. B., Udupi, V. R., & Patil, S. B. (2021). Identification of plant diseases using convolutional neural networks. *International Journal of Information Technology*, 13(6), 2461-2470.
16. Kabir, M. M., Ohi, A. Q., & Mridha, M. F. (2021). A multi-plant disease diagnosis method using convolutional neural network. *Computer Vision and Machine Learning in Agriculture*, 99-111.

17. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. Computers and electronics in agriculture, 147, 70-90.
18. Kundu, N., Rani, G., Dhaka, V. S., Gupta, K., Nayak, S. C., Verma, S., ... & Woźniak, M. (2021). IoT and interpretable machine learning based framework for disease prediction in pearl millet. Sensors, 21(16), 5386.
19. Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in plant science, 7, 1419.
20. Nevavuori, P., Narra, N., & Lipping, T. (2019). Crop yield prediction with deep convolutional neural networks. Computers and electronics in agriculture, 163, 104859.
21. Patil, B. V., & Patil, P. S. (2021). Computational method for Cotton Plant disease detection of crop management using deep learning and internet of things platforms. In Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020 (pp. 875-885). Springer Singapore.
22. Team, O.T.R. GoogleNet. 2021. Available online: <https://iq.opengenus.org/googlenet/> (accessed on 13 December 2021).
23. Upadhyay, S. K., & Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. International Journal of Information Technology, 1-15.
24. Yang, K., Zhong, W., & Li, F. (2020). Leaf segmentation and classification with a complicated background using deep learning. Agronomy, 10(11), 1721.

Figures

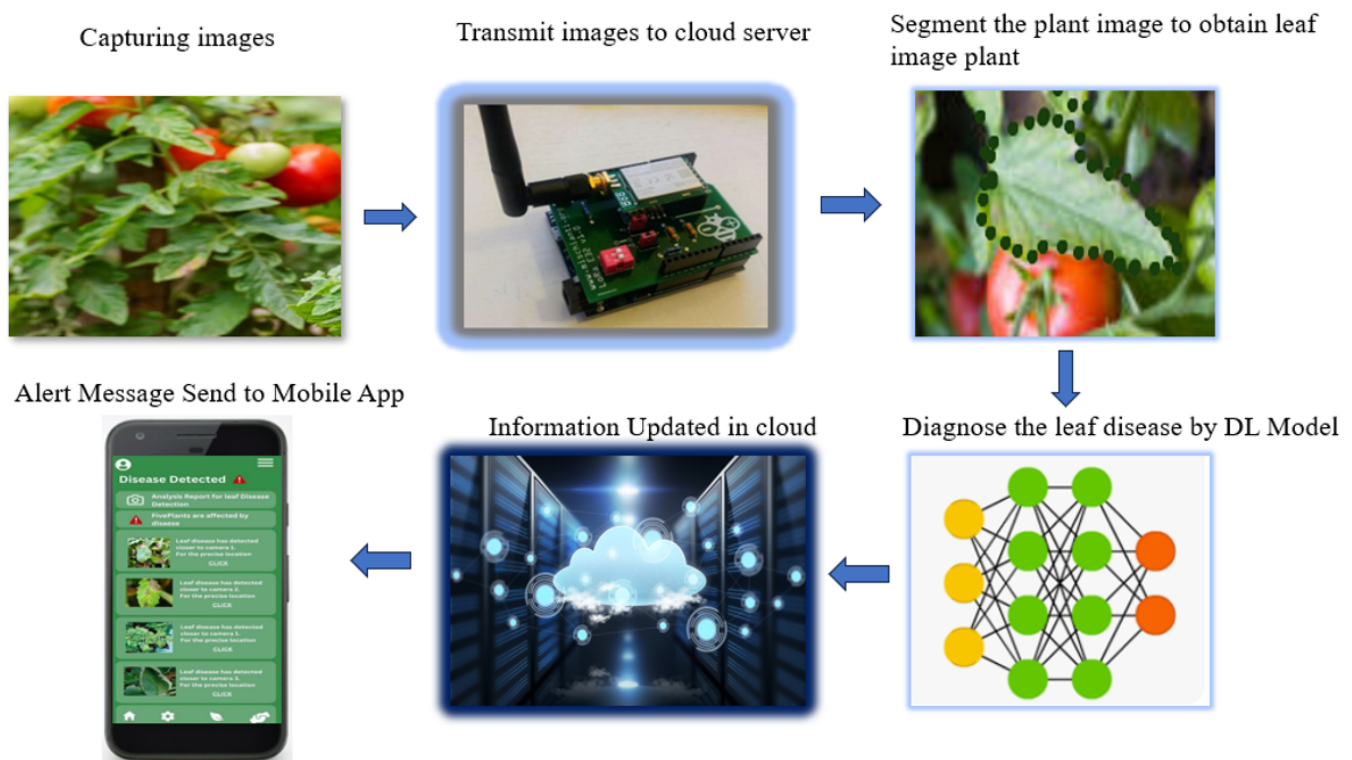


Figure 1

System Flow Block Diagram

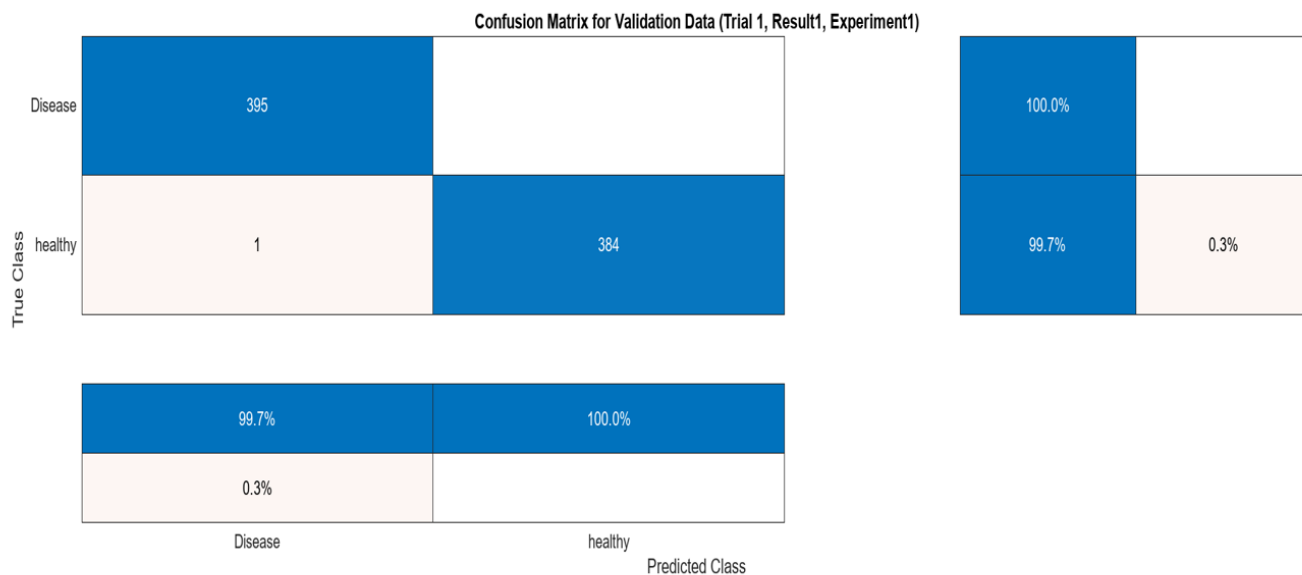
				
2.(a)	2.(b)	2.(c)	2.(d)	2.(e)
				
2.(f)	2.(g)	2.(h)	2.(i)	2.(j)

Figure 2

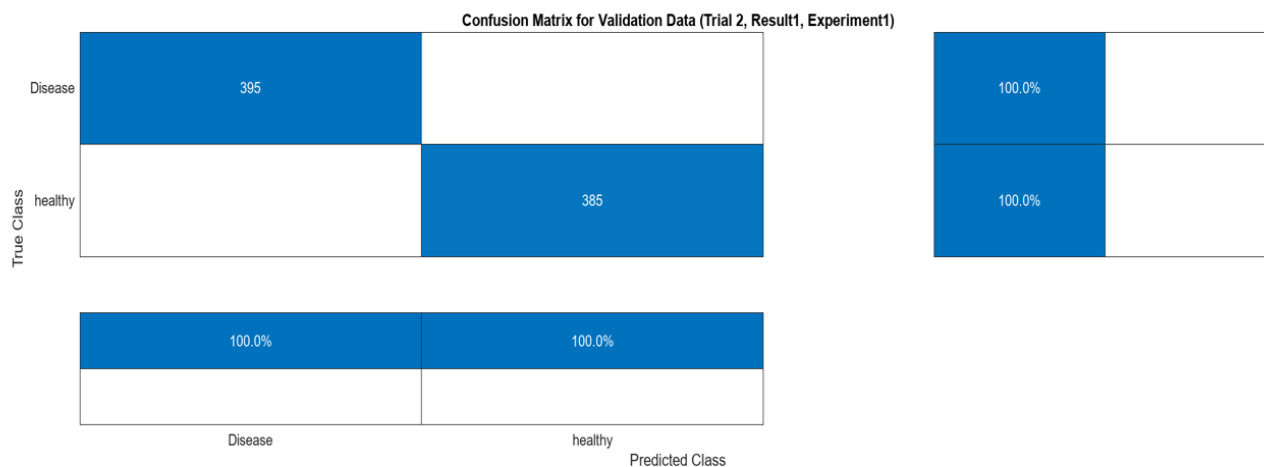
Sample leaf image with disease for (a) Healthy Leaf (b) Early Blight (c) Late Blight (d) Septoria Leaf Spot (e) Leaf Mold (f) Spider Mites (g) Target Spot (h) Tomato Mosaic Virus (i) Tomato Yellow Leaf Curl Virus (j) Bacterial Spot.

Experiment Details					Hyperparameters	Metrics			
Trial	Status	Actions	Progress	Elapsed Time	myInitialLearn...	Training Accur...	Training Loss	Validation Acc...	Validation Loss
1	Complete (...)		100.0%	6 hr 6 min 49 sec	0.0010	100.0000	0.0007	99.8718	0.0030
2	Complete (...)		100.0%	6 hr 6 min 19 sec	0.0100	100.0000	5.2486e-6	100.0000	0.0002
3	Complete (...)		100.0%	6 hr 8 min 44 sec	0.1000	98.4375	0.0660	98.5897	0.0303

a



b



c

Figure 3

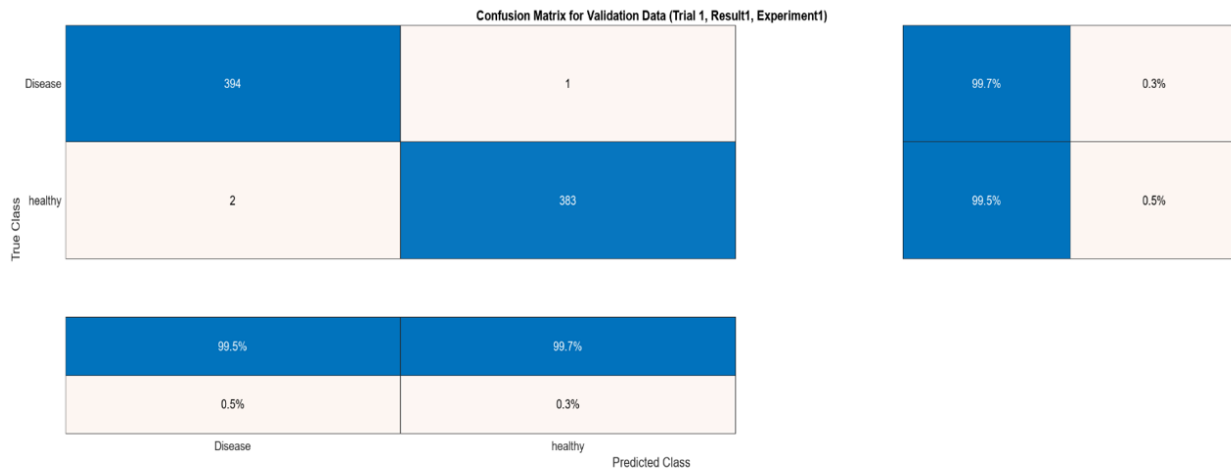
(a) Experiment result of ResNet 50 with various learning rate

(b) Confusion Matrix of ResNet 50 with learning rate 0.001

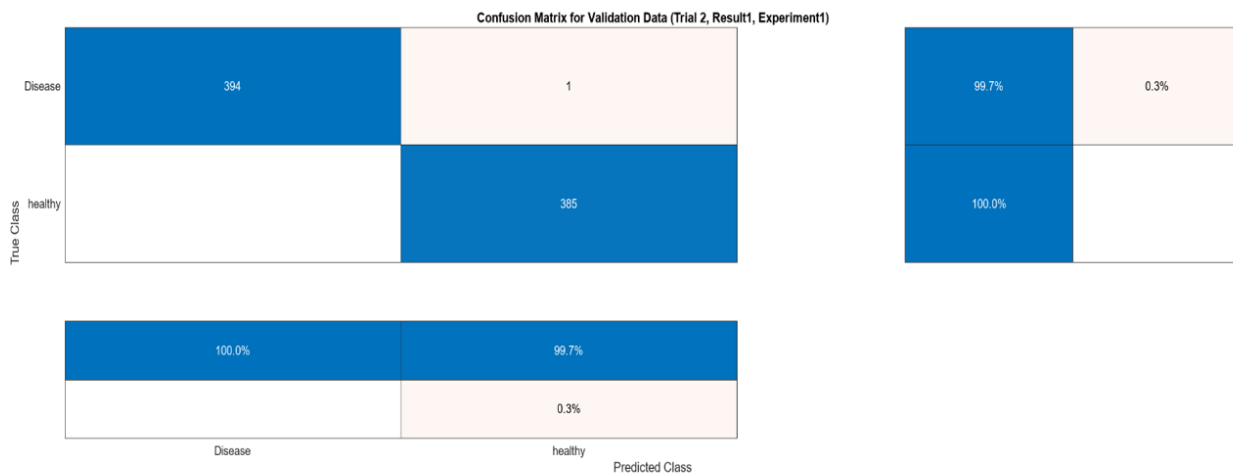
(c) ResNet 50 Confusion Matrix with 0.01 learning rate

Experiment Details						Hyperparameters	Metrics			
Trial	Status	Actions	Progress	Execution Envi...	Elapsed Time	myInitialLearn...	Training Accur...	Training Loss	Validation Acc...	Validation Loss
1	Complete [...]		100.0%	Single CPU	2 hr 4 min 35 sec	0.0010	100.0000	0.0006	99.6154	0.0189
2	Complete [...]		100.0%	Single CPU	2 hr 4 min 44 sec	0.0100	100.0000	5.2697e-6	99.8718	0.0026
3	Complete (T...		0.4%	Single CPU	0 hr 3 min 4 sec	0.1000	51.5625	NaN	50.6410	NaN

a



b



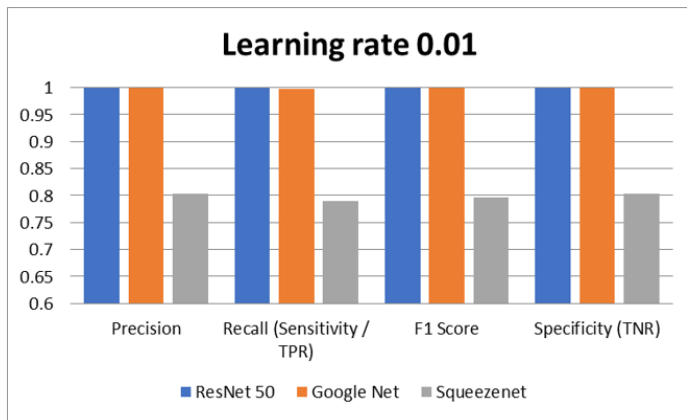
c

Figure 4

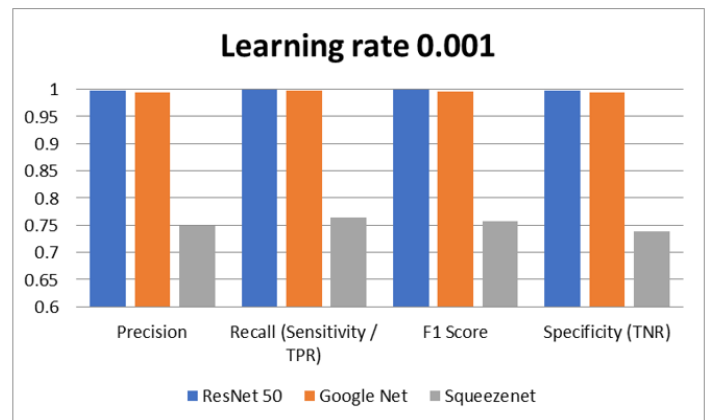
(a) Experiment result of GoogLeNet with various learning rate

(b) GoogLeNet Confusion Matrix with learning rate 0.001

(c) GoogLeNet Confusion Matrix with learning rate 0.01



a



b

Figure 5

(a) Performance Comparison at learning rate 0.01 & Figure 5(b) Performance Comparison at learning rate 0.001

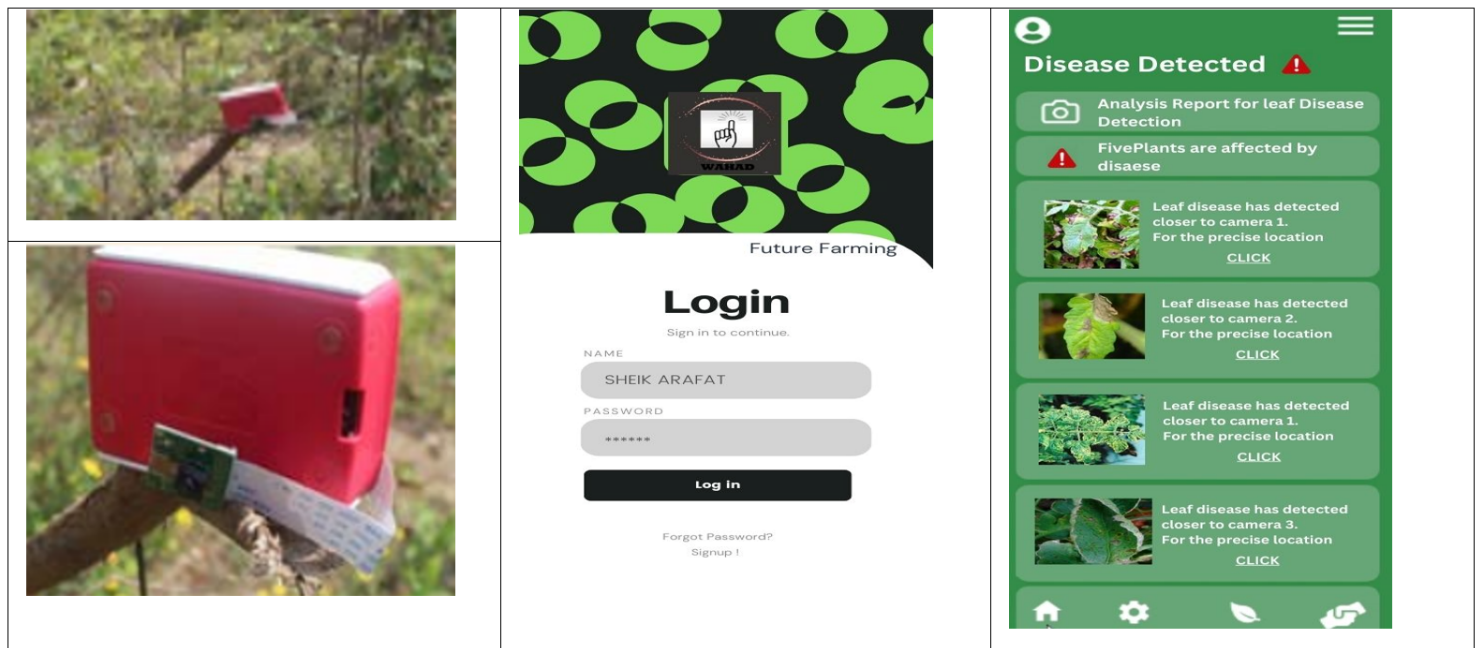


Figure 6

- a. Implemented Device in a tomato farm
- b. Login Page of the Mobile application
- c. Tomato leaf disease alert Page in Mobile app