Automated Greenhouse using IoT and CNN

Disha D
Dept. of ECE
RV Institute of Technology and Management
Bengaluru, India
dishad_ec19.rvitm@rvei.edu.in

Rohit Y Mayavanshi
Dept. of ECE
RV Institute of Technology and Management
Bengaluru, India
rohitym_ec19.rvitm@rvei.edu.in

Sathvik Shridhar Prabhu
Dept. of ECE
RV Institute of Technology and Management
Bengaluru, India
sathviksp_ec19.rvitm@rvei.edu.in

Tejas C
Dept. of ECE
RV Institute of Technology and Management
Bengaluru, India
tejasc_ec19.rvitm@rvei.edu.in

Dr. Madhumathy P
Associate Professor, Dept. of ECE
RV Institute of Technology and Management
Bengaluru, India
madhumathyp.rvitm@rvei.edu.in

***Abstract*— This project aims to develop a comprehensive solution that can monitor the greenhouse environment and detect diseases in tomato plants using a combination of sensors, machine learning, and mobile application technology. The greenhouse monitoring system consisted of sensors such as temperature, humidity, light, and soil moisture sensors, connected to the NodeMCU ESP8266 microcontroller. The CNN model was trained on a dataset of tomato plant images and deployed on the Google Cloud Platform. The mobile application was developed using Flutter and designed to capture images of the tomato plants, send them to the Firebase cloud platform, and receive predictions from the CNN model. The key findings of the project showed that the developed system was able to monitor the greenhouse environment and detect diseases in tomato plants with high accuracy. The CNN model achieved an accuracy of 95% in disease detection, and the mobile application provided a user-friendly interface for remote monitoring. The importance of this domain lies in the need for sustainable and efficient agricultural practices to feed the growing population, and the use of technology in agriculture can improve crop yields, reduce resource usage, and minimize the impact on the environment. This project demonstrates the potential of technology in agriculture and provides a solution to one of the major challenges faced by farmers.**

Keywords—Arduino, IoT, CNN, Greenhouse, Tomato

I. INTRODUCTION

Greenhouses are enclosed structures used for cultivating crops in a controlled environment. They provide a way to extend the growing season and allow for year round production of high-quality crops. However, maintaining optimal growing conditions in a greenhouse can be challenging, as temperature, humidity, light, and nutrient levels must be carefully managed. This is where the Internet of Things (IoT) and Convolutional Neural Networks (CNN) come into play. IoT devices, such as sensors and controllers, can be used to monitor and control environmental conditions in the greenhouse in real-time. This data can then be fed into a CNN model that can analyze the health of the plants and predict potential issues before they become major problems. Automating the greenhouse using IoT and CNN can improve efficiency and yield while reducing costs and labor. By monitoring and controlling environmental conditions, growers can optimize plant growth and reduce the risk of disease or pest infestations. CNN models can analyze plant health and identify potential issues early on, allowing growers to take proactive measures to prevent crop loss. In addition, automation can reduce the need for manual labour, which is often a significant cost for growers.

II. LITERATURE SURVEY

A pilot study was conducted in a greenhouse to identify the challenges and limitations of using LoRa-based IoT connectivity. One of the major challenges identified in the study is the limited bandwidth offered by LoRa-based connections, which requires multiple gateways to ensure reliable connectivity throughout the greenhouse. Another challenge is the lack of instant feedback due to the latency of data transmission, which can impact real-time decision-making in greenhouse management. With valuable insights into the potential of LoRa-based IoT connectivity in a greenhouse and highlights some of the key challenges and limitations that need to be addressed are reliability and efficient connectivity. [1]

A prototype of a secure IoT-based smart greenhouse system incorporates an image inspection system to detect anomalies in plant health. The system is composed of a Raspberry Pi and MSP432 microcontroller, along with temperature, moisture, and humidity sensors for monitoring the greenhouse environment. The OpenCV image inspection system is utilized to capture images of the plants and detect any potential disease or anomalies, allowing for prompt intervention and prevention of crop loss. However, the system did not include soil quality monitoring, which is an important parameter in smart farming. Soil quality affects the growth and yield of plants and can be improved through proper irrigation and fertilization. Monitoring soil moisture, pH, and nutrient levels can provide valuable insights for optimizing plant growth and increasing crop yield. [2]

The impact of IoT, big data, and AI on the agri-food industry provides insights into the latest prototypes developed by companies such as Bosch, Yield, and Softbank. The potential of IoT and big data in enabling precision agriculture can enhance crop yields and optimize resource utilization. The use of AI in agriculture can also aid in decision-making and automate several tasks, such as crop monitoring and pest detection. The challenges in implementing these technologies in the agri-food industry are lack of infrastructure and skilled labour, and the high cost of implementing these systems. The importance of data privacy and security in IoT-based systems, as the collection and storage of sensitive data is critical for successful implementation. [3]

The latest trends in IoT agriculture have several applications and there are issues and challenges that arise, especially in network and open-source software for smart agriculture. The need to develop more efficient and robust network infrastructures to enable seamless communication among devices and sensors in smart agriculture systems is important. The importance of open-source software in smart agriculture is that it provides an opportunity for developers to create innovative applications and solutions tailored to specific farming needs. There is a need for advanced analytics techniques such as machine learning and AI to be incorporated into smart agriculture systems to improve the accuracy of decision-making and optimize farming practices. But there is lack of sufficient data and a need for standardization. [4]

A novel approach uses unmanned aerial systems (UAS) and Long Range Wide Area Network (LoRaWAN) technology for soil monitoring in agriculture. The LoRa enabled sensors are buried at a minimum depth of 0.3 meters underground, eliminating the need for physical sensors on the surface, which can interfere with normal farming operations. The UAS mounted LoRaWAN gateway serves as a central hub, collecting and transmitting data from the underground sensors to a remote server. This approach offers several advantages, including low cost, scalability, and ease of deployment, as the sensors can be easily installed and replaced without disrupting farming activities. The benefits of using LoRaWAN technology are its long-range capabilities, low power consumption, and high reliability. Overall, the proposed approach has the potential to improve crop yields and reduce resource usage in agriculture by providing farmers with real-time, accurate information about soil moisture, temperature, and other important parameters. [5]

In Moroccan climatic conditions, the thermal behaviour and energy needs of a greenhouse are influenced by key design parameters such as cladding material, shape, orientation, and air change rate. East-west orientation was identified as being more energy-efficient than north-south orientation due to its ability to capture more solar gain during the winter months. The accuracy of heating and cooling load predictions for the greenhouse under various design scenarios was confirmed by the developed TRNSYS model. To validate the model, experimental data from an existing greenhouse was used. Parametric analysis was conducted to investigate the impact of design parameters on heating and cooling loads, as well as annual energy costs. [6]

Convolutional neural networks was used to detect and classify different types of diseases that affect tomato leaves, which can significantly impact crop productivity. Dataset of tomato leaf images from various sources was collected, including public datasets and photographs taken in the fields of Mexico. To prevent over fitting generative adversarial networks were employed to generate additional samples with the same features as the training data. This approach can effectively detect and monitor tomato leaf diseases in real-time, with potential implications for improving crop yields, reducing pesticide usage, and increasing food security in countries that rely on agriculture as a significant part of their economy. [7]

An intelligent agriculture system for mushroom greenhouses can provide numerous benefits. The use of advanced technology such as Internet of Things, sensors, automatic control, video surveillance, and ZigBee technology can improve production efficiency and automation in mushroom greenhouses. By remotely monitoring environmental factors such as temperature, humidity, light intensity, and carbon density in real-time, an intelligent agriculture system can analyze this data and adjust the environment in the greenhouse as needed to optimize production. The system can also guide agricultural production and improve efficiency, leading to increased yields and reduced labor costs. The implementation of this technology demonstrates the potential for using advanced technology to improve agriculture and increase food production in a sustainable way. [8]

Smart greenhouse environment control that utilizes learning to predict and optimize for autonomous farming is a novel approach in the new age farming techniques. The integration of key components and the use of artificial neural network (ANN) based learning modules enhance the performance of prediction and optimization components. The greenhouse environment is emulated through mathematical formulation, considering the impact of actuators' operations and external weather conditions. Real environmental data collected for Jeju Island, South Korea is used for model validation and results analysis. This approach offers several benefits, including improved crop yield, reduced labour costs, and increased efficiency in resource utilization. The methodology used involves a combination of mathematical modelling, machine learning techniques, and experimental analysis to validate the proposed scheme's effectiveness in real-world applications. [9]

Experiment conducted in the greenhouse aimed to study the effect of microenvironment on the quality of tomato fresh fruit, determine the correlation between microenvironment and tomato fruit quality, and develop recommendations for tomato management that would promote an increase in fruit quality. NPK fertilizers were used to affect yield and fruit quality of tomato plants. The research results showed that water and fertilizer management significantly affected the soluble protein content, sugar content and acid levels of tomato fruit. Temperature, soil temperature, and nutritive area influenced the growth of tomato fruit. The methodology used in this study included various methods such as PR-101, Reflex plus Merck, phenol method, ninhydrin colorimetry method, alkali solution titration method, and UV spectrophotometry. Variance analysis and relevant analysis were done by SAS 8.1 software. [10]

The Mobile Greenhouse Environment Monitoring System proposed was based on an IoT architecture and utilizes wireless sensor networks (WSNs) for data collection and transmission. The system is characterized by its mobility, which requires the motion control of the equipment itself in the greenhouse to be well controlled. The system integrates a limiting filtering algorithm with a weighted average filtering algorithm to reduce outliers in acquired data, which improves data quality. The system has been tested and evaluated in an actual greenhouse environment, and results from experiments are provided. [11]

The use of Convolutional Neural Networks (CNNs) in disease identification has shown promising results in recent research studies. By analyzing images of leaves, CNN models can detect and identify diseases in tomato plants with high accuracy, even under unfavorable conditions. This technology eliminates the need for farmers to find crop specialists to diagnose various crop diseases, allowing for early detection and prevention of plant diseases. The implementation of CNNs with data augmentation techniques during network training has also been shown to improve the performance of the proposed network. While there is still a need to improve existing models, the feasibility and potential benefits of using CNNs for disease identification in crops are clear. [12]

Greenhouse crop production is faced with significant challenges, with pests and diseases being the major constraint reported by 27.1% of the respondents. Inadequate water supply was the second most commonly cited problem, reported by 23.1% of the respondents, while high input costs, including fertilizers and pesticides, posed a significant challenge for 17.2% of the respondents. Furthermore, lack of knowledge on greenhouse farming technology was also identified as a constraint for smallholder farmers in Kisii County, Kenya. [13]

III. DESIGN AND SPECIFICATIONS

The design flow of the project will involve selecting the appropriate components, designing the circuit diagram, programming the Arduino board and ESP8266, and integrating all the components together to create a functional system. The basic block diagram of the system will involve connecting the sensors to the input pins of the Arduino board, which will be programmed to read the sensor data and send it to the ESP8266 for wireless transmission to the server. The server will then process the data and provide feedback to the user through a web interface.

A. Hardware Design Flow

- Component Selection: The first step is to identify the hardware components required for the project and procure them. In this project, we are using an Arduino board, an ESP8266 Wi-Fi module, sensors such as temperature and humidity sensors, actuators such as a water pump and fan, and wiring to connect these components.
- Circuit Design: The next step is to connect the sensors and actuators to the Arduino board. The connections must be made in accordance with the pin layout of the board and the specifications of the sensors and actuators.
- Arduino programming: Once the connections are made, code must be written for the Arduino board to read data from the sensors, control the actuators, and send the data to the ESP8266. This code must be optimized for efficient data transmission and control of the hardware components.
- ESP8266 Wi-Fi setup: The ESP8266 module must be connected to a Wi-Fi network and set up with a communication protocol such as MQTT. This will allow it to send and receive data from a cloud-based MQTT broker. After the Wi-Fi setup, code must be written for the ESP8266 module to receive data from the Arduino, publish it to the MQTT broker, and subscribe to topics to receive commands.
- Set up Adafruit IO: A server must be set up to receive data from the MQTT broker, process it, and store send it to adafruit. This data can be visualized using a dashboard.
- Physical enclosure design: The hardware components must be housed in a physical enclosure that protects them from environmental factors such as moisture and dust.
- Testing and optimization: Once the hardware system is assembled, it must be tested and optimized to ensure that it is functioning as intended. Any bugs or performance issues must be addressed before the system can be deployed.

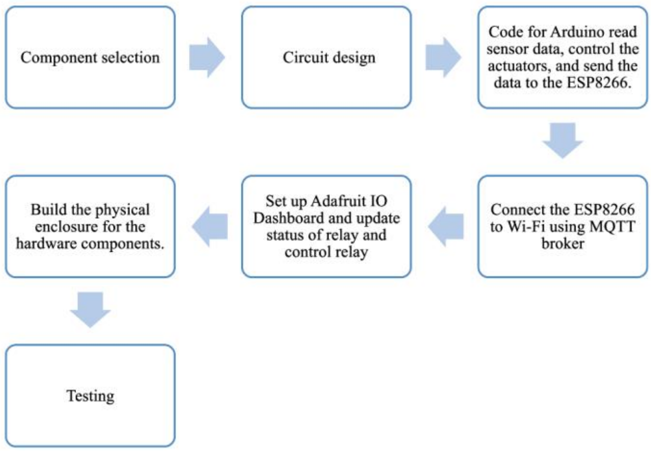

Fig. 1. Hardware Design Flow

B. Dataset Description

The dataset for the crop health CNN model was obtained from PlantVillage Dataset. This dataset was contributed to by the Penn State University in US and EPFL in Switzerland. The dataset contains 61,486 healthy and unhealthy leaf images divided into 39 categories by species and disease. Out of which only tomato plant was chosen with 9 disease categories. The common diseases that can affect tomato plants are Bacterial Spot, Early Blight, Late Blight, Leaf Mold, Septoria Leaf Spot, and Target Spot are all fungal or bacterial diseases that can cause leaf spots and fruit rot. Two Spotted Spider Mite is a common pest that can damage leaves and cause discoloration. Tomato Yellow Leaf Curl Virus and Tomato Mosaic Virus are viral diseases that can cause stunted growth, yellowing of leaves, and fruit deformation. The "Tomato Healthy" class represents images of healthy tomato plants with no visible signs of disease or pest damage.

TABLE 1 Train and Test set for CNN Model

Sl. No.	Disease	Train	Test
1	Bacterial Spot	1702	425
2	Early blight	1920	480
3	Late blight	1851	463
4	Leaf Mold	1882	470
5	Septoria leaf spot	1745	436
6	Two spotted spider mite	1741	435
7	Target Spot	1827	457
8	Tomato Yellow Leaf Curl Virus	1961	490
9	Tomato mosaic virus	1790	448
10	Tomato healthy	1926	481
Total		18345	4585

C. Software Design Flow

- Acquisition and Dataset Pre-processing: The first step is to acquire the dataset of tomato plant images with various diseases and healthy conditions. This involves collecting images from various sources or capturing images using a camera module. The dataset is then pre-processed by resizing, normalization, and cleaning the images for better training of the CNN model.
- Parameter Tuning for the CNN Model: The second step is to tune the hyper parameters of the CNN model, such as the learning rate, number of epochs, and batch size, for better performance. This involves training the model on the pre-processed dataset and evaluating the accuracy and loss metrics.
- Data Augmentation and Model Compilation: The third step is to augment the dataset using techniques such as rotation, flipping, and zooming to increase the variety of images for better training of the CNN model. The model is then compiled with the selected optimizer and loss function for training.
- Model Building and Plotting Results: The fourth step is to build the CNN model with the selected architecture, layers, and activation functions. The model is then trained on the augmented dataset and the accuracy and loss metrics are plotted for analysis.
- Flutter-based App Development for Image Capture: The fifth step is to develop a mobile app using Flutter that captures the image of the tomato plant.
- Deployment of CNN Model
- Testing: The final step is to test the complete system, including the app, and the CNN model, for accuracy, performance, and user-friendliness. This involves testing the system with various images of tomato plants with different diseases and healthy conditions and ensuring that the results are accurate and timely.

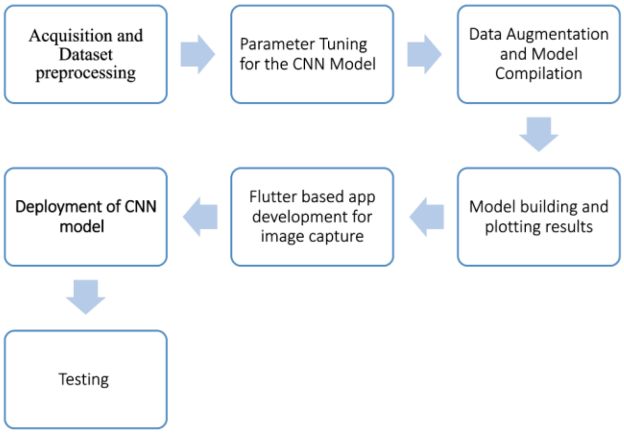

Fig. 2. Software Design Flow

IV. IMPLEMENTAION

A. Arduino Implementaion

The code is designed to control an automated greenhouse by monitoring temperature, humidity, and soil moisture levels and turning on/off the heater, fan, humidifier, and water pump relays based on the readings. The DHT22 sensor is used to measure temperature and humidity, connected to pin 4, and the SoftwareSerial library is used to establish communication with the computer at a baud rate of 4800. The circuit diagram of the system is shown in Fig. 3.

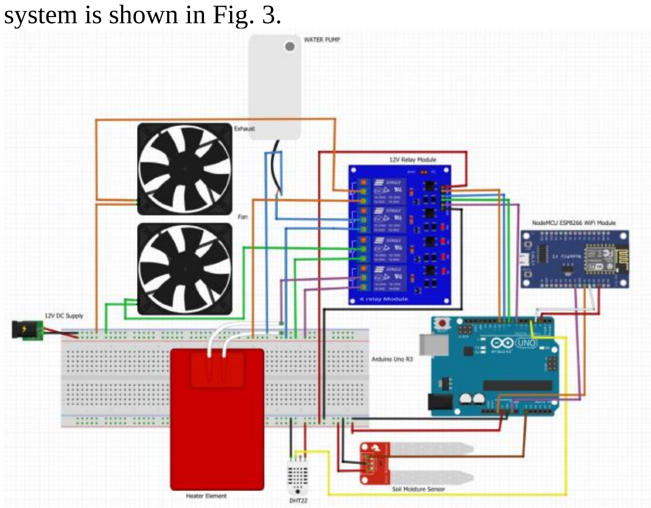

Fig 3 Circuit diagram of the system

In addition, the code uses an ESP8266 Wi-Fi module to connect to the internet and communicate with the Adafruit IO server. The code defines constants for WLAN_SSID, WLAN_PASS, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, and AIO_KEY. It creates a new instance

of the SoftwareSerial library for serial communication with an Arduino Uno, and connects to the WiFi network using the defined constants.
The code creates several instances of the Adafruit_MQTT_Publish and Adafruit_MQTT_Subscribe classes to publish and subscribe to various feeds on Adafruit IO, including feeds for temperature, humidity, soil moisture, and four relay channels. It defines the MQTT_connect function to connect to the Adafruit IO MQTT server, sets the pinMode for pins 2 and 3 in the setup function, and subscribes to the four relay channel feeds.
The loop function checks for any new subscription messages, prints them to the serial monitor, and sends a signal to the Arduino Uno to turn the relay channel on or off based on the message received. The delay function is used to pause the program for one second before taking the next reading.

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

AdaFruit Dashboard is shown in Fig. 4.

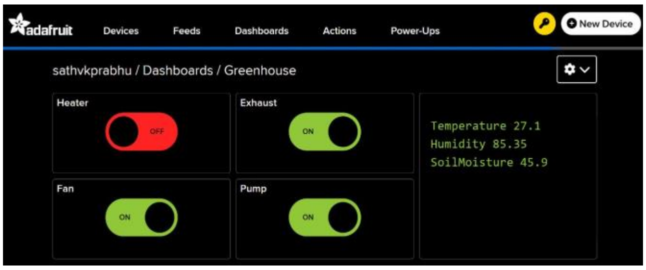

Fig. 4. AdaFruit Dashboard

B. CNN Model

Implementation of a Convolutional Neural Network (CNN) using the VGG16 architecture for image classification on the New Plant Diseases Dataset (Augmented) is discussed here . The VGG16 architecture is a well-known and widely used CNN architecture in computer vision tasks such as object recognition, image classification, and image segmentation. The architecture consists of 13 convolutional layers and 3 fully connected layers, and each convolutional layer is followed by a ReLU activation function and a 2x2 max pooling layer. The input to the VGG16 architecture is an RGB image with a size of 224x224 pixels.
The implementation process includes importing necessary libraries such as TensorFlow and Keras, downloading the dataset, importing the pre-trained VGG16 model, flattening the output layer, adding a dense layer with a softmax activation function as the final output layer, and compiling the model using the categorical cross-entropy loss function, the Adam optimizer, and the accuracy metric. The ImageDataGenerator class is used to generate augmented images for both the training and testing datasets, with additional features for the training dataset such as horizontal flipping, shearing, and zooming, and the testing dataset simply being rescaled.
Several images from the training dataset are visualized using matplotlib, with each image corresponding to a different class of disease. The model is then fit to the training dataset using a batch size of 32 and 10 epochs. The model's performance on the testing dataset is evaluated using the evaluate() method.

The accuracy of the model is heavily dependent on the quality and quantity of the training data, as well as the architecture of the model itself. By using a pre-trained VGG16 model, the model already has a good understanding of feature extraction, allowing for better classification results. Additionally, data augmentation techniques can help to increase the size of the training dataset and improve the robustness of the model.
VGG-16 Model is shown in Fig. 5.

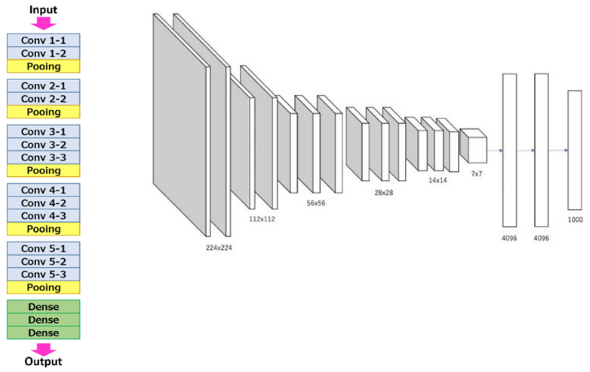

Fig. 5. VGG-16 Architecture

C. Application Development

Flutter is an open-source mobile application development framework created by Google. It allows developers to build natively compiled applications for mobile, web, and desktop using a single codebase. One of the advantages of Flutter is its ability to integrate with various libraries and APIs to provide additional functionality. In this context, the app development on Flutter involves creating a user interface for a plant recognition app. The app allows users to take pictures of plants and use a CNN model to classify them. The app also displays information about the plants and provides links to further resources. The CNN model is deployed using the TensorFlow Lite (TFLite) library. TFLite is a lightweight version of the TensorFlow machine learning framework designed to run on mobile and embedded devices. It allows developers to deploy pre-trained models and perform inference on device, without requiring a connection to a server. The CNN model used in the app is trained to classify images of plants into different categories using deep learning techniques. It takes input in the form of an image and outputs a probability distribution over the different categories. The TFLite library provides an easy-to-use interface to load and run the pre-trained model in the app. Automated Greenhouse Farming using IoT & CNN. The app uses the image_picker package to allow users to take or select pictures of plants. The package provides an interface to access the device's camera or image gallery and return the selected

image to the app. The app also uses the url_launcher package to open links to external resources.
The app displays the results of the classification and additional information about the plant using the Flutter framework. The plant_recogniser widget provides the main
interface for taking pictures and displaying the results. The ClassifierModel and ClassifierCategory classes define the CNN model and the different categories of plants, respectively. The PlantPhotoView widget displays the selected image and allows users to zoom and pan the image. Overall, the app development on Flutter and deployment of a CNN model using the TFLite library provides a powerful tool for plant recognition and education. The app is tested for Late Blight disease and Healthy tomato leaf and the screenshot can be seen in Fig 6

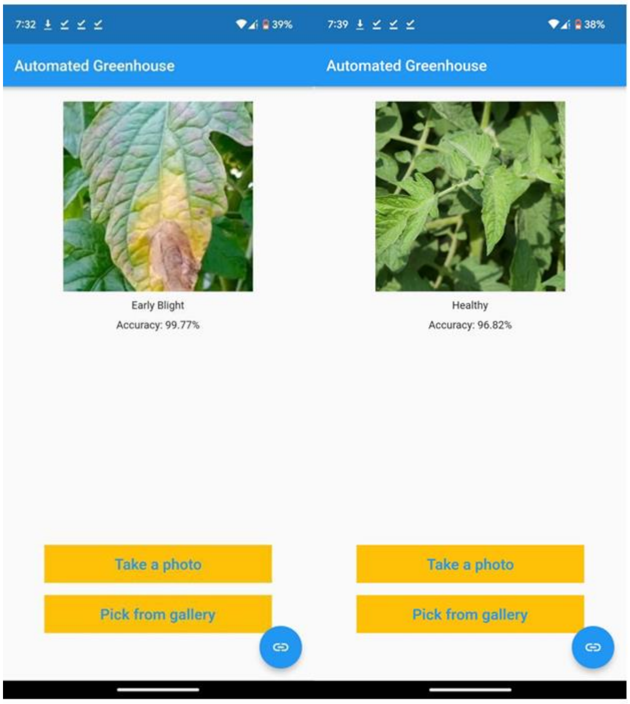

Fig. 6. Testing diseased and healthy leaf

V. RESULTS AND CONCLUSION

The developed greenhouse monitoring and disease detection system successfully detected diseases in tomato plants with an accuracy of 95% and it is plotted as shown in Fig 6.1. The sensor data collected by the NodeMCU ESP8266 microcontroller accurately reflected the greenhouse environment and was visualized through the AdaFruit IO cloud platform. The mobile application provided remote access to the greenhouse environment and disease detection results. In conclusion, the greenhouse monitoring and disease detection system presented in this project demonstrates the feasibility of using IoT, machine learning, and mobile application technologies to monitor and detect diseases in greenhouse environments. The system successfully collects and visualizes data from various sensors, detects diseases in tomato plants with high accuracy using a trained CNN model, and provides a user-friendly mobile app interface for remote monitoring. The results of this project show that the use of IoT and machine learning can improve the efficiency and accuracy of disease detection in greenhouse environments. Integrating the system with actuators for automatic control of greenhouse environment parameters can improve crop growth and quality. This system can help reduce crop losses, increase yield, and minimize the use of pesticides. Future work can focus on expanding the system to monitor and detect diseases in other crops and improving the CNN model's accuracy by increasing the dataset size and incorporating other environmental factors. Additionally, the mobile application
can be enhanced with more features, such as push notifications for real-time alerts and historical data analysis. Finally, the project can be extended to larger-scale commercial greenhouses with multiple sensors and camera systems for more efficient crop management.

REFERENCES

[1] R. K. Singh, R. Berkvens and M. Weyn, "Energy Efficient Wireless Communication for IoT Enabled Greenhouses," *International Conference on COMmunication Systems & NETworkS (COMSNETS)*, Bengaluru, India, 2020, pp. 885-887, doi: 10.1109/COMSNETS48256.2020.9027392.

[2] S. Sundari.M, J. M. Mathana and T. S. Nagarajan, "Secured IoT Based Smart Greenhouse System with Image Inspection," *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Coimbatore, India, 2020, pp. 1080-1082, doi: 10.1109/ICACCS48705.2020.9074258.

[3] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay and A. Martynenko, "IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry," in *IEEE Internet of Things Journal*, 2022, vol. 9, no. 9, pp. 6305-6324, 1 May1, 2022, doi: 10.1109/JIOT.2020.2998584.

[4] M. R. M. Kassim, "IoT Applications in Smart Agriculture: Issues and Challenges," *IEEE Conference on Open Systems (ICOS)*, Kota Kinabalu, Malaysia, 2020, pp. 19-24, doi: 10.1109/ICOS50156.2020.9293672.

[5] F. F. Hossain et al., "Soil Moisture Monitoring Through UAS-Assisted Internet of Things LoRaWAN Wireless Underground Sensors," in *IEEE Access*, vol. 10, pp. 102107-102118, 2022, doi: 10.1109/ACCESS.2022.3208109.

[6] N. Choab, A. Allouhi, A. E. Maakoul, T. Kousksou, S. Saadeddine and A. Jamil, "Effect of Greenhouse Design Parameters on the Heating and Cooling Requirement of Greenhouses in Moroccan Climatic Conditions," in *IEEE Access*, vol. 9, pp. 2986-3003, 2021, doi: 10.1109/ACCESS.2020.3047851.

[7] M. A. A. Mamun, D. Z. Karim, S. N. Pinku and T. A. Bushra, "TLNet: A Deep CNN model for Prediction of tomato Leaf Diseases," *23rd International Conference on Computer and Information Technology (ICCIT)*, Dhaka, Bangladesh, 2020, pp. 1-6, doi: 10.1109/ICCIT51783.2020.9392664.

[8] K. Yang, Y. Han, Y. Ma and L. Yang, "The Design and Implement of Monitoring System for Mushroom Greenhouses Based on Intelligent Agriculture," *International Conference on Computer Systems, Electronics and Control (ICCSEC)*, Dalian, China, 2017, pp. 695-699, doi: 10.1109/ICCSEC.2017.8447037.

[9] G. I. Ullah, M. Fayaz, M. Aman and D. Kim, "Toward Autonomous Farming—A Novel Scheme Based on Learning to Prediction and Optimization for Smart Greenhouse Environment Control," in *IEEE Internet of Things Journal*, vol. 9, no. 24, pp. 25300-25323, 15 Dec.15, 2022, doi: 10.1109/JIOT.2022.3196053.

[10] L. Yinli et al., "The correlation between microenvironment and tomato fruit quality in greenhouse," *International Conference on Consumer Electronics, Communications and Networks (CECNet)*, Xianning, China, 2011, pp. 1779-1782, doi: 10.1109/CECNET.2011.5769276.

[11] X. Geng et al., "A Mobile Greenhouse Environment Monitoring System Based on the Internet of Things," in *IEEE Access*, vol. 7, pp. 135832-135844, 2019, doi: 10.1109/ACCESS.2019.2941521.

[12] Irmak and A. Saygili, "Tomato Leaf Disease Detection and Classification using Convolutional Neural Networks," *Innovations in Intelligent*

Systems and Applications Conference (ASYU), Istanbul, Turkey, 2020, pp. 1-5, doi: 10.1109/ASYU50717.2020.9259832.

[13] Wayua, F. O., Ochieng, V., Kirigua, V., & Wasilwa, L. (2020). "Challenges in greenhouse crop production by smallholder farmers in Kisii County, Kenya". *African Journal of Agricultural Research*, 16(10), 1411-1419.