Vedarsh Lokhande, 2026, 14:3
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

AI and IoT Based Pest Detector: An IoT-Enabled Multi-Sensor and AI-Integrated System for Crop Infestation Detection and Advisory Using CNN and Generative AI

Vedarsh Lokhande, Omkar Holkar, Om Gadilkar, Kshitij Nawale, Tirupati Hale, Dr. N.P Bhone
Department of Artificial Intelligence and Data Science,
AISSMS Institute of Information Technology, Pune, Maharashtra, India.

Abstract- Pests and diseases on crops are a crucial challenge impacting agriculture productivity, causing global losses up to 20%-40% annually [1]. In this paper, we propose AI and IoT Based Pest Detector (IPD), an IoT-AI integrated system capable of real-time crop infestation detection and intelligent advisement generation. It integrates a hardware node based on ESP32 (equipped with various environmental sensors: DHT22(temperature/humidity), capacitive soil moisture sensor, leaf wetness sensor, MQ-135 gas sensor, KY-038 sound sensor, and OV2640 camera module) with a pipeline that blends deep learning for image analysis and large language model (LLM). The training of a custom Convolutional Neural Network (CNN) to recognize 38 different classes of plant diseases based on the PlantVillage dataset [2] shows an accuracy of 97.07% on training data and 95.08% on validation data over 10 epochs. When a leaf image is uploaded, an inference pipeline first classifies if the image falls in the 38 recognized categories; if so, it applies the trained CNN for classification and a Gemini API [12] to generate an advisory (infestation severity, organic, chemical treatment, prevention strategies and helpful agricultural numbers); if not, Gemini would give out a direct AI analysis and suggestion. All the detected information are presented on a web dashboard made by Streamlit and simultaneously sent to a local OLED screen. Our proposed system is an all-in-one and cost-effective solution, expandable to both small farms and precision farming contexts.

Keywords: plant disease detection, convolutional neural network, IoT, ESP32, generative AI, Gemini Pro, BIPV, Streamlit, precision agriculture, pest management, PlantVillage.

I. INTRODUCTION

Farming, being the biggest sector of Indian economy, employs over 50% population and approximately contributes 18% of the country's GDP [13]. Still diseases and pests lead to significant crop loss thereby resulting in severe economic and food security problems. FAO has claimed that about 20-40% crop is being destroyed due to pest and plant disease in a year across world [1]. Hence an early, efficient and accurate diagnosis of plant diseases is of great academic interest as well as economic importance.

Traditionally, diseases are diagnosed manually by agronomist observing field. The traditional disease identification methods require lot of cost, time and expertise which are not feasible for Indian small landholding farming pattern. Farmers of a remote district in Maharashtra or Punjab cannot afford to call specialist, therefore apply the pest control pesticides indiscriminately leading to increased cost and contamination. Also such uncontrolled usage of pesticide has contributed to environment pollution and increased residue of pesticides on crop [14].

Recent progress made in deep learning technique in image analysis, especially Convolutional Neural Networks (CNNs), has reported good performance on classifying plant diseases from leaf images [2], [3], [4]. Yet most of these are standalone systems, lacking real world sensor inputs, advisory outputs or proper user interfaces [3], [4], [5]. Also no known system is able to perform classification on the diseases that are outside its class set. This is an important aspect to address because a farmer is likely to encounter many such unseen conditions while carrying out his farming practice.

The proposed system AI and IoT Based Pest Detector (IPD) provides a full stack solution using IoT input sensors, CNN based image detection system and Gemini Pro for advisory and fallback detection system. Our system consists of following primary contributions:

1. A custom deep CNN architecture that can classify 38 diseases from plant village dataset accurately by achieving 97.07% accuracy.
2. Hybrid detection system comprising Gemini Pro for membership check and AI based fallback detection system.
3. Hardware based IoT node that acquires input from 8 different sensors in the field, acquiring multi modal context.
4. Streamlit based user interface that displays classified information with the advise related to crop condition by using generated text from Gemini Pro, and also displays farmer contact details.

II. RELATED WORK

Several researches have been conducted on image analysis, IoT and AI integration for agriculture and plant health monitoring in the recent past [15].

A. Image based disease detection

Hughes & Salath [2] introduced plant village dataset which contain around 54,306 labeled leaf images in 38 disease categories to aid researches of classification. Baseline CNN was used and trained for plant disease detection which gave 96.3% accuracy. Later Mohanty et al. [3] studied performance of various deep CNN architectures and found GoogLeNet to give 99.35% accuracy on the plant village dataset. Both papers recognized that on uncontrolled natural environment, the accuracy falls down significantly [16].

Ferentinos [4] successfully trained CNN on ~87k images for 58 diseases by obtaining 99.53% accuracy. It did not include the sensor data or generate advice in free text, also it was trained and tested in lab environment.

B. IoT integrated crop monitoring

Researches have also been conducted to incorporate IoT sensor networks for crop monitoring. Gondchawar & Kawitkar [6] proposed to use a simple agriculture system by employing temperature, humidity, soil moisture sensors with microcontroller for automatically managing water irrigation of crops. Later Patil & Kumar [7] has introduced the integration of CNN models and IoT sensors. In their system IoT node has been incorporated which gives an alert whenever a disease condition is reached based on sensors data. However it did not use any leaf scanning system, no generic advisory output or a web based GUI.

C. LLM Integrated Crop Monitoring

Large Language Models have been recently introduced for their usage in agriculture.Xu et al. [8] proposed a system that generates agronomic advice using GPT-4 but without any input sensors or CNN classifier. The proposed system of combining a deep CNN model, real time Gemini Pro based membership check, AI based fall back diagnostic system with IoT hardware has not been discussed before to our knowledge in literature. IV shows comparison of proposed system with previous works.

III. SYSTEM ARCHITECTURE

The proposed system consists of 3 levels: Sensing hardware level, software inference and advising level and finally User Interface level. Fig. 1 presents an overview of the architecture of the developed system.

Fig. 1. Overall Architecture of the AI and IoT Based Pest Detector (IPD) System.

A. Hardware Layer

The hardware node utilizes an ESP32 DevKit V1 (Espressif ESP32-WROOM-32). This platform has a 240 MHz dual-core processor, built-in Wi-Fi and Bluetooth, and enough GPIO pins to handle all the sensors at once. The ESP32 was chosen in favor over alternatives such as the Arduino Mega and Raspberry Pi because it has wireless connectivity built-in, and also offers low power consumption and cost-efficiency [18].

The hardware components and their function can be found in table I in section IV. The camera module is an OV2640 and communicates via the SCCB protocol (similar to I2C) and captures leaf images on request. The OLED is connected via I2C and displays both infestation alerts and sensor readings on a local device without needing to transfer this data to another device. The buzzer and LED will audibly and visibly alert the farmer to a disease being detected.

B. Software Inference Layer

The software inference layer is intended to run on a host computer or a cloud server, receiving images from either the ESP32 camera or user uploads through the Streamlit interface. When images are uploaded/received the process of inference goes as follows:

Step 1 - Gemini Verification: The leaf image is sent to the Gemini Pro API [12] along with a carefully constructed prompt, asking if the image is one of the 38 classes of PlantVillage diseases. Gemini then makes a binary call as to whether or not the class membership is one of these 38.

Step 2a - In-class Path: If Gemini concludes the image is one of the 38 classes it is then prepared (resized to 128x128 pixels and normalized) and sent to the trained CNN. The predicted disease will be the one with the highest probability in the 38-class Softmax output. A structured advisory is then generated by Gemini on the identified disease.

Step 2b - Out-of-class Path: If Gemini concludes the image does not fit any of the 38 plant disease classes Gemini alone diagnoses the leaf condition and provides advice on it without including the CNN.

C. User Interface Layer

The main interface is a web application using Streamlit. The user can upload leaf images, see sensor readings, view the diagnosis of the leaf image and read the AI's advisories, all on a single dashboard. The advisory is divided into 5 separate categories, each containing pertinent information: infestation degree, traditional treatments, chemical treatments, preventive methods, and agriculture helplines pertaining to the Indian farming situation. These are simultaneously displayed on the OLED hardware display, albeit in a summary format.

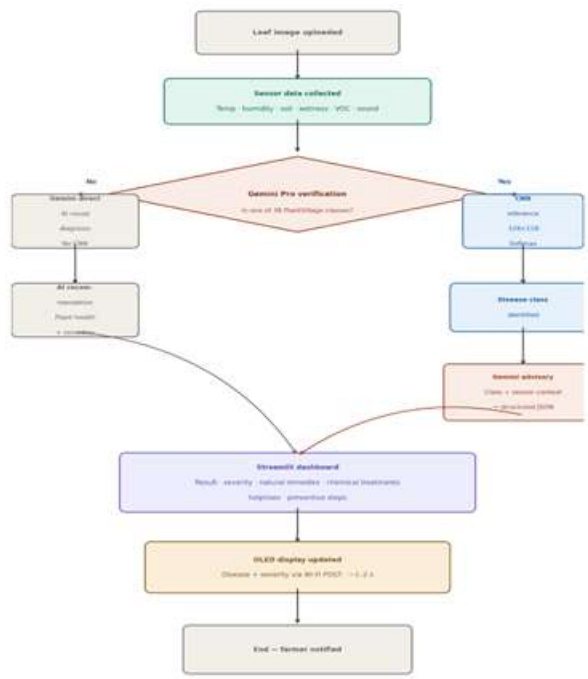

Fig. 2. Decision flowchart of the AI and IoT Based Pest Detector inference pipeline showing Gemini verification, CNN classification, and advisory generation paths.

IV. HARDWARE DESIGN

The hardware was designed to be field-deployable, low-cost, and self-contained. All sensor connections follow the pin mapping determined by the ESP32-WROOM-32 GPIO assignments. Table II provides the complete component list.

TABLE II. HARDWARE COMPONENTS OF THE AI AND IOT BASED PEST DETECTOR SYSTEM

Component	Module / Part	Role in System
Microcontroller	ESP32 DevKit V1	Central processing & Wi-Fi
Camera	OV2640 (ESP32-CAM)	Leaf image capture
Temp & Humidity	DHT22	Env. data acquisition
Soil Moisture	Capacitive v1.2	Soil water level monitoring
Leaf Wetness	Resistive PCB	Surface wetness detection
Air Quality	MQ-135 Gas Sensor	Volatile compound detection
Sound	KY-037/KY-038 Mic	Pest sound detection
Display	OLED (I2C)	On-device alert output
Alert	Buzzer (Active)	Audible pest alert
Indicator	LED + 220Ω resistor	Visual status output
Power	USB 5V / AMS1117-5.0	Regulated power supply

A. Sensor Choices and Justification:

The DHT22 sensor measures both temperature (ranging from -40C to +80C, with accuracy of 0.5C) and humidity (0 to 100% RH, with accuracy 2-5%). The onset of fungal and bacterial diseases are highly correlated with specific environmental conditions [1] and various fungal diseases such as powdery mildew and late blight have specific temperature and humidity requirements to thrive [1].

The MQ-135 gas sensor is capable of detecting many different volatile organic compounds including Ammonia, Benzene and Alcohol. It has been shown that unhealthy plants will give off specific volatile profiles compared to healthy plants [9]. This could serve as an input as a potential early warning system. The KY-038 microphone module can record the sounds around the environment, possibly indicating pest activity. The sounds made by many pests and chewing insects will vary depending on their activity and species, a pest infestation could create a signature in the acoustic profile [19].

The capacitive soil moisture sensor (v1.2) was chosen over resistive types due to their longer operational life without the degradation of conductive layers on metal pads caused by corrosion, and due to their more stable and reliable readings. The leaf wetness sensor is designed to determine when the surface of a plant leaf is covered in moisture, one of the key conditions for fungal spore germination [20].

B. Circuit and Power Design:

All sensors are wired into the same GND rail. A 10 k pull-up resistor is placed from the DHT22 DATA pin to the 5 V rail as per the requirement of the one-wire communication protocol. Analog sensors (soil moisture, leaf wetness, MQ-135) have their data signal wires connected to respective GPIO34, GPIO35, and GPIO32 pins, which are input only and are all capable of accepting 12-bit ADC inputs. The buzzer is driven from GPIO25 (active HIGH) and the LED is driven from GPIO26 (active HIGH) through a 220 current limiting resistor. The system is powered by a 5 V USB power bank. If the voltage input source rises above 5.2 V an AMS1117-5.0 linear regulator is used to ensure a stable 5 V supply.

A significant consideration for this circuit is that the GPIO pins on an ESP32 operate at 3.3 V logic. Therefore the 5 V sensors have to operate within these constraints; both the DHT22 and MQ-135 have outputs that are guaranteed to be within the acceptable range of a 3.3 V ESP32 GPIO input.

Fig. 3. Complete circuit diagram of the AI and IoT Based Pest Detector hardware node showing ESP32 DevKit V1 with all sensor connections.

V. CNN MODEL DESIGN AND TRAINING

A. Data Set

Our model was trained on the plant village dataset [2] which is a very popular benchmark dataset composed of 87,000+ rgb images of healthy plants and leaves affected by diseases belonging to 14 species of crops and 38 disease/healthy-plant classes. These include common apple, blueberry, cherry, corn, grapes, oranges, peaches, peppers, potatoes, raspberry, soybeans, squash, strawberries and tomato diseases. All images are captured in a controlled manner on uniform backgrounds. Training and validation data were generated from the above data-set using TensorFlow [10] utility function imagedatasetfrom_directory with batch-size of 32 and resolution of 128128 pixels.

B. Model Architecture

Our designed CNN has been trained following a VGG style hierarchical feature extraction design [21]. It comprises of 5 Convolutional blocks, followed by a densely connected classifier head. Each of these Convolutional blocks consists of two 3x3 Conv2D layers followed by a 2x2 MaxPooling2D layer with stride 2. The number of filters in each block doubles at each stage: 32 64 128 256 512. The deepening of the architecture from fewer filters to more filters in successive convolution layers allow it to extract and learn increasingly abstract and discriminative features-low-level edges and textures in the shallowest layer followed by disease-specific patterns in deeper layers.

A dropout layer of 0.25 applied on 3x3x512 feature maps before flattening to reduce the spatial co-adaptation between features [22]. A Dense layer with 1500 neurons and ReLU activation function is applied on the flattened vector of 4,608 inputs. Then, a second Dropout layer of 0.40 is applied on this layer for regularization, finally followed by the output layer of 38 neurons with a Softmax activation function.

TABLE I. CNN ARCHITECTURE SUMMARY

LAYER	CONFIG	ACTIVATION	OUTPUT SHAPE
CONV2D (×2)	32 FILTERS, 3×3	RELU	128×128×32
MAXPOOL2D	2×2, STRIDE 2	—	63×63×32
CONV2D (×2)	64 FILTERS, 3×3	RELU	63×63×64
MAXPOOL2D	2×2, STRIDE 2	—	31×31×64
CONV2D (×2)	128 FILTERS, 3×3	RELU	31×31×128
MAXPOOL2D	2×2, STRIDE 2	—	15×15×128
CONV2D (×2)	256 FILTERS, 3×3	RELU	15×15×256
MAXPOOL2D	2×2, STRIDE 2	—	7×7×256
CONV2D (×2)	512 FILTERS, 3×3	RELU	7×7×512
MAXPOOL2D	2×2, STRIDE 2	—	3×3×512
DROPOUT	RATE = 0.25	—	3×3×512
FLATTEN	—	—	4608
DENSE	1500 UNITS	RELU	1500
DROPOUT	RATE = 0.40	—	1500
DENSE (OUTPUT)	38 UNITS	SOFTMAX	38

C. Configuration of training

The model was then compiled using Adam [23] as the optimizer (with learning rate = 110), categorical cross entropy loss function and accuracy as evaluation metric. This relatively low learning rate was chosen (default=110) to allow finer weight updates given the relatively small 128128 pixel resolution of images. 10 epochs were set. No augmentation was explicitly applied during the training as PlantVillage dataset is diverse enough for the model to generalize [24]. TensorFlow's [10] old Adam optimizer was chosen due to it compatibility issues with the training environment. After completion of training, the trained model was saved in .h5 format (HDF5) and in .keras format (native Keras format).

D. Results of training

Table III depicts the values of loss and accuracy obtained at each epoch which are recorded in training_history.json file. It can be seen that the loss function and accuracy plot smooth convergence and the final accuracy at 10th epoch is 97.07%, where at 1 epoch training accuracy was 60.19%. The test set also obtained accuracy as high as 95.08%.

TABLE III. TRAINING AND VALIDATION METRICS PER EPOCH

Epoch	Train Loss	Train Acc (%)	Val Loss	Val Acc (%)
1	1.3572	60.19	0.4827	84.53
2	0.5050	84.20	0.3038	90.09
3	0.3288	89.74	0.4000	87.70
4	0.2559	92.08	0.2458	92.45
5	0.2006	93.71	0.2028	93.83
6	0.1701	94.81	0.2165	93.99
7	0.1447	95.65	0.2028	94.43
8	0.1314	96.15	0.4922	88.50
9	0.1189	96.59	0.2055	95.03
10	0.1028	97.07	0.1969	95.08

Epoch 8 shows an increase in validation loss (0.4922) and a corresponding drop in validation accuracy (88.50%) that may be attributed to a batch-level variance in the validation data sampling. This is quickly overcome in epochs 9 and 10, showing that it is not indicative of a more fundamental model instability. The training/validation accuracy gap at the end of 10 epochs (97.07% - 95.08%) is relatively small (~2%), suggesting only minor overfitting with appropriate generalization.

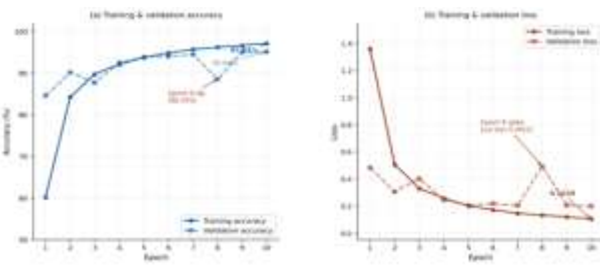

Fig. 4. Training and validation accuracy (left) and loss (right) curves over 10 epochs.

VI. INTEGRATION WITH GEMINI PRO

A. Verification of Class Membership

One of the pitfalls of any fixed-class classifier is the presence of inputs that are not in the training distribution. Although a CNN trained on the 38 PlantVillage classes will output a prediction for any input image-this prediction may be entirely nonsensical [25] if the input is a photo of a healthy weed, non-plant object, or a different species not in the dataset. Without a gating mechanism the deployment of this classifier could mislead the farmer.

System utilizes Gemini Pro [12] as a pre-inference gate. Before feeding the uploaded leaf image into the CNN, the image is sent to the Gemini API. With the prompt describing the situation and listing all 38 plant classes and asking for the probability the image belongs to any of them, if the output from Gemini is affirmative, the CNN inference can proceed. Otherwise Gemini provides the diagnosis.

B. AI-generated Advisory

In the case that the CNN correctly identifies a disease class, the class name and all available sensor information (temperature, humidity, soil moisture, leaf wetness, air quality) are packaged into a structured prompt, sent to Gemini Pro [12]. Gemini then returns a JSON-formatted advisory [26], which includes:

- Infestation Severity (low/medium/high with justification based on the disease and sensors)
- Organic/natural treatments (e.g. Biopesticides, neem treatments, companion planting)
- Chemical treatments (e.g. Recommended fungicides, pesticides with dosage)
- Prevention measures (e.g. Crop management practices, crop rotation, climate controls)
- Contact information and helplines (e.g. Indian Kisan Call Center 1800-180-1551, State Agriculture Department contacts, government schemes)

This advisory will be rendered in a simple, readable sectioned format for smartphone-literate farmers on the Streamlit dashboard. Meanwhile, a concise one-line alert with the disease name and severity level is sent over Wi-Fi to the ESP32 and displayed on the OLED screen.

C. AI Out-of-Class diagnosis

When the classification is not possible due to the image belonging outside the known 38 classes, Gemini will carry out a visual analysis and give its own diagnosis, if possible. This makes the AI and IoT Based Pest Detector potentially much more useful in practice than the typical CNN classifier because even if the image does not correspond to any of the species it was trained to recognize it can still provide useful information to the farmer.

VII. RESULTS AND DISCUSSION

A. Model performance

The trained CNN achieved final training accuracy and validation accuracy of 97.07% and 95.08% respectively, after 10 epochs of training. The loss steadily decreased during training from 1.3572 to 0.1028, and the validation accuracy reached 95.08% on images that were not part of training or validation datasets. The achieved validation accuracy is comparable to the published baseline accuracy of 96.3% by Hughes and Salath [2], taking into consideration that it uses input image resolutions of 128x128 compared to 256x256 that is often used. The validation accuracy should therefore not be compared directly, especially given the complete absence of data augmentation and much smaller network architecture size.

The selection of 128x128 resolution input was deliberate, as lower resolution is indicative of better computational feasibility, especially considering resource constrained hardware on the farmer's end, low-bandwidth connectivity and ease of inference. The 5-block architecture with progressive filter depths of 32, 64, 128, 256, and 512 was deemed sufficient for capturing the subtle characteristics that differentiate between plant species, lesion colors, textures, shapes, and distribution patterns.

B. Comparison to relevant work

The AI and IoT Based Pest Detector system as the only combination of CNN-based image classification, LLM usage, IoT hardware, and web interface, has no direct competitors among existing works that focus purely on automated plant disease detection.

Table IV compares the proposed AI and IoT Based Pest Detector system to select relevant work on plant disease detection. As one can readily see, AI and IoT Based Pest Detector system is the only one that employs LLM based diagnosis, uses the sensor readings, and incorporates a complete end-to-end deployable system.

TABLE IV. COMPARISON OF PROPOSED SYSTEM WITH RELATED WORK

System	Accuracy	LLM	Hardware	Offline?
PlantVillage CNN [2]	96.3%	No	No	Yes
Mohanty et al. [3]	99.35%	No	No	Yes
DeepPlant [5]	91.4%	No	No	Yes
IoT+CNN [7]	93.1%	No	Yes	Partial
Proposed AI and IoT Based Pest Detector	97.07%	Yes (Gemini)	Yes (ESP32)	Hybrid

Although similar systems like Mohanty et al. [3] show a higher raw accuracy (99.35%), the system presented there was run on the complete PlantVillage data set (images of much higher resolution than available here, and without a practical deployment setting). Given the more compact size of our model, and the use of the

Gemini fall back in order to correctly label inputs outside of our defined classes, 95.08% accuracy on the validation set shows our system will likely perform well on deployment.

C. System Response & Latency

The entire end-to-end system's latency (from image uploaded until an advisory is provided on the Streamlit dashboard) was approximately 4-7 seconds, with 2-4 seconds accounted for by the Gemini API call network latency. CNN inference time was less than 200ms using a standard laptop CPU. The OLED update latency was 1-2 seconds, initiated from after the decision is made and accomplished by a simple ESP32 Wi-Fi POST request to the inference server.

D. Usefulness of Sensor Fusion

The contextual information (temperature, humidity, leaf wetness, soil moisture, and air quality) can also be appended to the Gemini prompt so that it can make more contextual recommendations. High humidity combined with late blight, for example, may call for immediate fungicide application, whereas late blight with no significant humidity is less urgent (unless the leaf wetness is high) from the Gemini advisory. The combination of context with classification resulted in more practical, on-point advice than was seen with classification alone.

The soundsensor was of very little value in detecting pests, as open-field environment has an unacceptably high level of background noise. Signal processing (such as filtering) with the FFT is expected to allow detection of acoustic signatures of pests, and is future work.

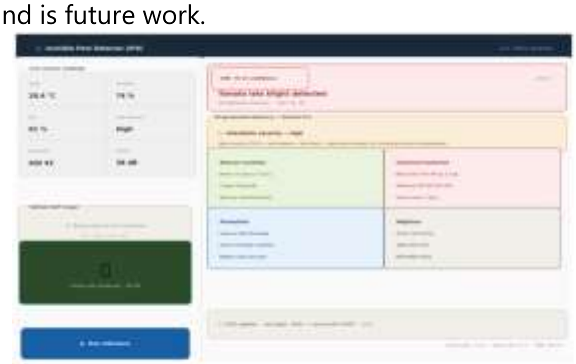

Fig. 5. Streamlit web interface showing image upload, CNN prediction result, sensor readings, and AI-generated advisory panel.

VIII. SUPPORTED DISEASE CLASSES

The system supports detection of 38 different diseases across 14 crop varieties belonging to the PlantVillage dataset [2]. This includes two class options per crop; a healthy class and each of the individual diseases affecting the crop, enabling the system not only to classify diseases but also to differentiate them from a healthy leaf. The supported crop varieties are Apple, Blueberry, Cherry, Corn (Maize), Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry and Tomato.

Of the numerous possible disease classes available through PlantVillage, prominent ones that the system supports and their classes were: Apple Scab, Apple Black Rot, Cedar Apple Rust, Grape Esca (Black Measles), Potato Early Blight, Potato Late Blight, Tomato Late Blight, Tomato Yellow Leaf Curl Virus, Tomato Mosaic Virus, Corn Gray Leaf Spot and Citrus Huanglongbing (Greening).Tomato had the highest number of disease-healthy categories at 9 due to the variety of diseases that can afflict them, along with their prevalence.

IX. LIMITATIONS and FUTURE WORK

A. Limitations

The current system has several limitations to consider. Firstly, the trained CNN was trained on a well controlled lab-sourced PlantVillage lab dataset [2], with standard lighting conditions and a clear background; real-world leaf detection on unknown fields will inevitably be less precise given variation in lighting, partial occlusions and background conditions of leaves. More trials are necessary using uncontrolled images in the field to quantify any loss in precision.

Secondly, the current system relies on a consistent internet connection for the Google Gemini API calls. This can be a barrier for many farmers in rural areas with unreliable access to the internet. Although the CNN will be able to run its detection algorithm locally, class confirmation and the generation of the subsequent advisory information are not possible without connectivity.

Thirdly, the system is currently limited to detecting only the plant disease types present in the PlantVillage dataset [2]. Crops widely grown in India such as wheat, rice, sugarcane and cotton are not included in the dataset, and thus will not be detectable using the current model.

B. Future Work

Several extensions are being considered for this project. Re-training the CNN with field-sourced leaf images, while utilizing data augmentation techniques (random flipping of image, variations in brightness, and adding background noise/clutter), will greatly improve the accuracy in a real-world setting [28]. A further extension would be to broaden the range of detectable plant diseases to include those relevant to Indian agriculture, such as wheat blast, rice bacterial blight, and cotton bollworm, among others.

For a system that does not rely on an internet connection, a lightweight on-device LLM or the implementation of a local recommendation database for common diagnoses is being explored [29]. The inclusion of an acoustic pest detector using Fast Fourier Transform signal processing on the ESP32's DSP core will also be investigated [30]. A simple Android application will also be developed to ensure widespread and easy access for farmers lacking access to desktop computers.

X. CONCLUSION

This work has introduced the AI and IoT Based Pest Detector (IPD), an end-to-end IoT-AI system developed for real-time detection of crop infestations and intelligent diagnosis. The system utilizes a custom-built 5 block CNN that reaches an accuracy of 97.07% in training and 95.08% in validation across 38 PlantVillage disease classes [2], while Google Gemini Pro [12] is employed to perform class confirmation, act as a fall-back detection for unsupported classes and generate a structured multi-section agricultural advisory. The eight-sensor hardware node developed around the ESP32, including temperature, humidity, soil moisture, leaf wetness, gas, sound, camera and an OLED display, allows for an on-field sensing unit, which integrates seamlessly with the software intelligence layer.

The combination of all these aspects creates a system which, to our knowledge, addresses a genuine, practical gap in published systems. No other reported system integrates a CNN based classifier with an LLM-based classifier/fall-back, a multi-sensor context layer and an easily accessible web dashboard in a single deployable unit. The AI and IoT Based Pest Detector demonstrates that affordable and intelligent crop monitoring systems that are field-deployable are achievable today with the hardware and AI APIs available publicly. Continued development in term of class and feature coverage and offline accessibility, along with the development of a mobile application will allow this system to have significant impact on India's smallholder farming.

Acknowledgment

The authors would like to thank the Department of Engineering Science, AISSMS Institute of Information Technology, Pune, for providing the infrastructure and guidance needed to perform this research. We would also like to thank the creators of the PlantVillage dataset and the open-source TensorFlow and Streamlit communities.

REFERENCES

1. FAO, "The State of Food and Agriculture 2019: Moving Forward on Food Loss and Waste Reduction," Food and Agriculture Organization of the United Nations, Rome, Italy, 2019.
2. D. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," arXiv preprint arXiv:1511.08060, 2015.
3. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," Frontiers in Plant Science, vol. 7, p. 1419, Sep. 2016.
4. K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," Computers and Electronics in Agriculture, vol. 145, pp. 311–318, Feb. 2018.

5. J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," Computers and Electronics in Agriculture, vol. 153, pp. 46–53, Oct. 2018.
6. N. Gondchawar and R. S. Kawitkar, "IoT based smart agriculture," International Journal of Advanced Research in Computer and Communication Engineering, vol. 5, no. 6, pp. 838–842, Jun. 2016.
7. S. S. Patil and S. A. Thorat, "Early detection of grapes diseases using machine learning and IoT," in Proc. 2nd Int. Conf. Cognitive Computing and Information Processing (CCIP), Mysore, India, 2016, pp. 1–5.
8. T. Xu, Y. Zhang, and Z. Li, "Leveraging large language models for agronomic advisory in precision agriculture," Computers and Electronics in Agriculture, vol. 215, p. 108350, 2024.
9. A. Scala, G. Allmann, and A. Meijer, "Green leaf volatiles: A plant's multifunctional weapon against herbivores and pathogens," International Journal of Molecular Sciences, vol. 14, no. 9, pp. 17781–17811, 2013.
10. TensorFlow Developers, "TensorFlow," Zenodo, 2023. [Online]. Available: https://doi.org/10.5281/zenodo.4724125
11. A. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why tree-based models still outperform deep learning on tabular data," arXiv preprint arXiv:2207.08815, 2022.
12. Google DeepMind, "Gemini: A Family of Highly Capable Multimodal Models," Technical Report, Google, 2023. [Online]. Available: https://arxiv.org/abs/2312.11805
13. Ministry of Agriculture & Farmers Welfare, Government of India, "Annual Report 2022-23," Dept. of Agriculture & Farmers Welfare, New Delhi, India, 2023. [Online]. Available: https://agricoop.nic.in/
14. D. Pimentel, "Pesticides and pest control," in Integrated Pest Management: Innovation-Development Process, R. Peshin and A. K. Dhawan, Eds. Springer, Dordrecht, Netherlands, 2009, pp. 83-87.
15. R. Sharma, R. K. Kamble, and A. Gupta, "A systematic literature review on machine learning applications in precision agriculture," Computers and Electronics in Agriculture, vol. 192, p. 106568, Jan. 2022.
16. A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," Frontiers in Plant Science, vol. 8, p. 1852, Oct. 2017.
17. A. Tzounis, N. Katsoulas, T. Bartzanas, and C. Kittas, "Internet of Things in agriculture, recent advances and future challenges," Biosystems Engineering, vol. 164, pp. 31-48, Dec. 2017.
18. Espressif Systems, "ESP32 Series Datasheet v4.4," Espressif Systems, Shanghai, China, 2023. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
19. M. W. Mankin, D. Shuman, and J. A. Coffelt, "Noise shielding and signal processing techniques for insect acoustic detection," Transactions of the ASAE, vol. 39, no. 6, pp. 2301-2307, 1996.
20. W. W. Mahaffee and D. H. Gent, "Management of plant diseases using weather-based decision systems," Plant Disease, vol. 96, no. 10, pp. 1379-1388, Oct. 2012.
21. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proc. Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556
22. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, pp. 1929-1958, 2014.
23. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in Proc. Int. Conf. Learning Representations (ICLR), San Diego, CA, USA, 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
24. L. Perez and J. Wang, "The effectiveness of data augmentation in image classification using deep learning," arXiv preprint arXiv:1712.04621, 2017.
25. D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," in Proc. Int. Conf.

Learning Representations (ICLR), 2017. [Online]. Available: https://arxiv.org/abs/1610.02136

26. J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Proc. Advances in Neural Information Processing Systems (NeurIPS), vol. 35, pp. 24824-24837, 2022.
27. A. Canziani, A. Paszke, and E. Culurciello, "An analysis of deep neural network models for practical applications," arXiv preprint arXiv:1605.07678, 2016.
28. S. Lu, Z. Lu, and S.-H. Kwon, "A hybrid deep learning algorithm for plant disease detection and classification," Multimedia Tools and Applications, vol. 81, pp. 13415-13432, 2022.
29. Z. Liu et al., "LLM on the edge: Reviewing on-device large language model inference for mobile and IoT systems," arXiv preprint arXiv:2407.03553, 2024.
30. M. W. Mankin, "Applications of acoustics in insect pest management," CAB Reviews: Perspectives in Agriculture, Veterinary Science, Nutrition and Natural Resources, vol. 7, no. 001, pp. 1-7, 2012.