© 2026 ADYPJIET, Volume 1(1), Online ISSN: 3049-1517, June 2026, pp. 38-49

Ajeenkya DY Patil Journal of Innovation in Engineering & Technology

Journal Homepage: https://www.adypsoe.in/adypjiet

AI-Powered Crop Monitoring and Management with Blockchain Traceability

[1]Shivam Suryawanshi,
[2]Abhijit Padghan,
[3]Sujal Nighot,
[4]Tejal Pawar
[5]Prof. Nalini Tiwari

[1-4] UG students, Ajeenkya D Y Patil School of Engineering, Pune, INDIA
[5]Assistant Professor, E & TC Dept, Ajeenkya D Y Patil School of Engineering, INDIA

Article History:
Received: 19/04/2026
Revised: 04/05/2026
Accepted: 20/06/2026

Abstract:
This paper presents a simple and compact system for the early detection of diseases in soybean crops. The system allows farmers to take a photo of a soybean leaf and upload it through a user-friendly web interface. The uploaded image is then processed by the backend, which analyzes its color and texture features to identify the condition of the leaf. Based on this analysis, the system classifies the leaf into one of three categories: healthy, affected by Bacterial Blight, or damaged by *Diabrotica speciosa*. After the leaf is classified, the result, along with its confidence score and the time of detection, is securely stored in a blockchain ledger. Each record is connected to the previous one using SHA-256 hashing, making the ledger tamper-resistant. If any stored record is modified, the chain is broken, making the change easy to detect. A React.js dashboard displays the classification result immediately after processing and allows users to verify that the stored records remain unchanged. To evaluate the system, a dataset of 800 labelled soybean leaf images was used. Out of these, 640 images were used to train the model, while the remaining 160 images were reserved for testing its performance. On the test dataset, the model achieved an accuracy of 89.66% and a weighted F1-score of 0.88**,** demonstrating reliable performance in classifying soybean leaf conditions. The system was also efficient, with each prediction taking less than 100 milliseconds on standard consumer hardware. To evaluate the security of the blockchain, 100 consecutive transactions were recorded, and all were stored successfully without any integrity issues. By combining AI-based disease detection with blockchain technology, the system provides farmers with

more than just a diagnosis. Each detection result is securely recorded in a tamper-resistant ledger, ensuring that any attempt to modify the stored information can be identified immediately. This makes the system both reliable for disease detection and trustworthy for maintaining secure records.

Keywords Support Vector Machine; Blockchain Traceability; Crop Disease Detection; Proof-of-Work; Precision Agriculture; Leaf Image Classification; Flask; React.js.

1.0 Introduction

More than 570 million households around the world depend on agriculture for their livelihood, and global food production will need to support an estimated 9.7 billion people by 2050. Despite this growing demand, crop monitoring in many regions still relies on traditional methods. Farmers usually inspect their fields manually, examine plant leaves, and identify diseases based on their experience. While this approach works well for small farms and familiar conditions, it becomes difficult to manage on a larger scale and may not always provide consistent results.

According to the Food and Agriculture Organization (FAO), pests and plant diseases that are not detected early are responsible for up to 40% of annual crop losses [1]. These losses highlight the importance of faster and more reliable disease detection. The challenge is becoming even greater due to climate change, which is altering pest populations and disease patterns, making them harder for farmers to predict and manage.

To address these challenges, machine learning has emerged as an effective solution for automatic crop disease detection. By analyzing images of plant leaves, machine learning models can accurately distinguish between healthy and diseased plants, allowing farmers to take timely action and reduce crop losses [2]. Among the available techniques, Support Vector Machines (SVMs) are particularly suitable because they perform well even with small datasets, are easy to interpret, and require relatively low computational resources. These advantages make SVMs a practical choice for agricultural applications where large labelled datasets and powerful computing systems are often unavailable. Recent studies have also shown that SVMs combined with carefully designed image features can achieve performance that is comparable to deep learning methods, especially when training data is limited [3].

While AI has made crop disease detection faster and more accurate, one major challenge still remains making sure that the prediction results are reliable and cannot be changed after they are generated. If these records can be modified, farmers, distributors, insurance companies, regulators, and consumers may hesitate to trust AI-based crop health reports. Therefore, it is important to have a secure system that can prove the data has remained unchanged over time.

Blockchain technology helps solve this problem by storing every prediction in a secure, cryptographically linked block within an append-only ledger. Because each block is connected to the previous one, any attempt to alter a stored record changes the entire chain, making tampering easy to detect. This ensures that the records are secure, transparent, and easy to verify [4]. Earlier studies have shown the benefits of blockchain for improving food supply chain traceability [5] and protecting agricultural data [7]. However, very few studies have integrated AI-based crop disease detection with blockchain-based record verification into a single working system. This research aims to address that gap by combining both technologies in one practical solution.

This paper presents an integrated solution called AI-Powered Crop Monitoring and Management with Blockchain Traceability. The proposed system combines an SVM-based soybean leaf disease classifier, a lightweight local Proof-of-Work (PoW) blockchain, and a React.js dashboard into a single web-based platform. Farmers can upload an image of a soybean leaf through a web browser, after which the system identifies whether the leaf is healthy, affected by Bacterial Blight, or damaged by *Diabrotica speciosa*. Each prediction is then securely stored on the blockchain, creating a permanent and tamper-resistant record that can be verified whenever needed.

The main contributions of this work are as follows. First, it provides a practical AI and blockchain framework that can be accessed using only a smartphone camera and a web browser. Second, it uses an SVM classifier based on a 296-dimensional handcrafted feature vector, achieving an accuracy of 89.66% for soybean leaf disease classification. Third, the proposed local Proof-of-Work blockchain maintained 100% chain integrity during testing with 100 consecutive transactions, demonstrating the reliability of the record-keeping process. Finally, the React.js dashboard enables users, even those without technical expertise, to view disease predictions and verify blockchain records in real time through an easy-to-use interface.

The rest of this paper is organized as follows. Section II reviews the existing literature and highlights the research gaps that motivated this work. Section III describes the proposed system architecture and the methodology used to develop the framework. Section IV presents the experimental setup and discusses the results. Finally, Section V concludes the paper and suggests possible directions for future research.

2.0 Literature Survey

A. AI-Based Crop Disease Detection

Research on AI-based plant disease detection has grown significantly over the years, with studies exploring different machine learning techniques and smart farming technologies. Assimakopoulos and Vassilakis examined the combined use of artificial intelligence, the Internet of Things (IoT), and blockchain in smart agriculture. Their work provided a strong conceptual foundation for developing integrated agricultural monitoring systems. Botero-Valencia et al. compared several machine learning approaches, including Support Vector Machines (SVM), Convolutional Neural Networks (CNN), and ensemble methods, for crop management applications. Their results showed that SVMs can perform effectively, particularly when only a limited amount of labelled training data is available. Miller et al. investigated the integration of IoT devices with AI-based sensing systems for field monitoring and demonstrated that connecting these technologies can provide timely and reliable information about crop health, helping farmers make better management decisions. Gutub [6] proposed a custom deep learning model, CoffNet**,** for detecting diseases in

© 2026 ADYPJIET, Volume 1(1), Online ISSN: 3049-1517, June 2026, pp. 38-49

coffee leaves. The model achieved an impressive accuracy of 99.89%**,** showing that deep learning can identify plant diseases from leaf images with very high accuracy. This study highlights the potential of image-based AI techniques for improving disease detection in agriculture. The study also acknowledged that models of that depth depend on large domain-specific datasets and offer no mechanism for verifying or securing prediction records. Both constraints directly informed the present work's decision to use SVM at the prototype stage.

B. Blockchain for Agricultural Traceability

Demestichas et al. [8] looked at how blockchain is used across agri-food traceability and found that distributed ledgers consistently make supply chain data more transparent and easier for consumers to trust. Yang et al. [5] surveyed combined blockchain-IoT deployments in agriculture and livestock contexts, concluding that the pairing is effective at producing crop records that are both verifiable and resistant to tampering. Ellahi et al. [7] focused on blockchain-enabled food supply chains and identified immutable logging and smart contracts as the features that do the most work when it comes to accountability. Demestichas and Peppes [4] reviewed the combined use of AI and blockchain, noting that despite their complementary strengths, end-to-end implementations that bring both together in a working system remain uncommon a gap the present work directly addresses. Salah et al. [12] used blockchain technology to improve transparency in the soybean supply chain. However, their system relied on manual data entry and did not include any AI-based method for automatically detecting or predicting the health of soybean crops.

C. IoT-Enabled Precision Farming

Kim and Lee [11] combined IoT sensors with blockchain technology on a Raspberry Pi to improve the traceability and quality monitoring of perishable products. Their study showed that sensor data can be securely stored using blockchain. Iftikhar and Khan [13] developed a low-cost smart sensor system for monitoring food storage conditions and demonstrated that affordable embedded hardware can be used effectively in agricultural applications. Similarly, Hossain et al. [9] integrated IoT sensors with blockchain technology for dairy product monitoring, creating a secure and tamper-resistant record of storage conditions for perishable goods.

TABLE I. COMPARATIVE ANALYSIS OF RELATED WORK

Ref.	Focus	AI Model	Blockchain	Dataset	Gap
[1]	AI for smart agriculture review	SVM, CNN, DL	Yes	Multiple crops	No prototype
[2]	ML in sustainable agriculture	CNN, SVM	No	Multiple	No blockchain
[3]	IoT & AI sensing review	CNN	No	Field sensors	No traceability
[4]	AI-Blockchain integration review	DL	Yes	Generic	No implementation

[5]	Blockchain-IoT for agriculture	None	Yes	Supply chain	No AI model
[6]	Deep learning for leaf disease	CNN (CoffNet)	No	Coffee leaves	No blockchain
[7]	Blockchain food supply chains	None	Yes	Food chain	No AI
Ours	**AI + Blockchain crop monitoring**	**SVM (RBF)**	**PoW**	**800 soybean**	**Full pipeline**

D. Research Gaps

The literature review shows three main gaps that this study addresses. First, most studies focus only on AI for crop disease detection or blockchain for data security, but very few combine both in one complete system [1][2][4]. Second, many AI-based systems need expensive hardware, large training datasets, or cloud computing, which are not practical for small farmers [3][6]. Third, only a few studies provide a simple web-based system with an easy-to-use dashboard where farmers can view disease results and check that the records stored in the blockchain have not been changed [4][5]. Table I compares the existing studies and shows how this work fills these gaps.

3.0 System Architecture and Methodology

The system has three main parts that work one after another: image collection and preparation, AI-based disease detection, and secure storage of records. A React.js interface helps users easily use all three parts of the system. The system is simple, lightweight, and does not need special hardware or complicated setup, making it easy to use in real-life farming.

A. Overview of the Proposed System

The farmer opens the application in a web browser, selects a photo of the plant leaf, and uploads it. The React frontend sends the image to the Python Flask backend using an HTTP POST request. The backend processes the image, extracts the required features, and sends them to the SVM model. The SVM then identifies the disease and returns the predicted disease name along with a confidence score. The result is then sent to the /add block endpoint, where it is stored as a new record in the local blockchain. The frontend receives the result and displays the detected disease, block hash, timestamp, and blockchain status on a single screen. The entire process, from uploading the image to storing the result in the blockchain, takes about 600–800 milliseconds on average.

B. Data Collection and Image Processing

The dataset contains 800 labelled images of soybean leaves collected from open-source repositories. It includes about 280 healthy leaves, 260 leaves affected by Bacterial Blight, and 260 leaves showing damage caused by *Diabrotica speciosa*. *Diabrotica speciosa* is a beetle that feeds on soybean leaves, creating many small holes in the leaf tissue. The AI model detects this leaf damage, not the beetle itself.

Each leaf image is resized to 128 × 128 pixels and converted into three color formats**:** RGB, HSV, and LAB**.** The images are then normalized to reduce the effect of different lighting conditions. To improve the performance of the AI model, the 640 training images are increased using data augmentation. During this process, the images are randomly flipped horizontally, rotated by up to ±15°**,** and their brightness is adjusted by up to ±20%**.** This helps the model learn from more image variations and improves its accuracy. These changes increase the variety of the training images by about 40%. This helps the model learn better from different types of images and reduces the chance of it simply memorizing the training data instead of learning to identify diseases correctly.

C. Feature Extraction

Before an image is given to the classifier, it is converted into a set of 296 numerical features**.** Out of these, 288 features are obtained from color histograms created using the HSV, LAB, and BGR color spaces. These features represent the color information in the leaf image and help the model identify different plant conditions more accurately. The remaining 8 features describe the texture of the leaf. These include the mean, standard deviation, skewness, and kurtosis of the grayscale pixel values. Since these features have different value ranges, Scikit-learn's Standard Scaler is used to standardize them before training. This makes all the features have a similar scale, which helps the model learn more accurately. Without standardization, features with larger values could affect the model more than they should, leading to less accurate disease detection.

D. SVM Model Training

The system uses a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel to classify the leaf images. The RBF kernel helps the model identify complex patterns and accurately distinguish between healthy and diseased leaves.

$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ …(1)

where γ governs how far each support vector's influence reaches. Through empirical tuning, C was fixed at 10 value and γ set to 'scale', resolving to 1 / (n_features × X.var ()) at runtime. The dataset of 800 images was divided into 640 images for training and 160 images for testing**.** The split was done carefully so that each group contained a similar number of healthy and diseased leaf images. During training, five-fold cross-validation was used to improve the model's performance and reduce overfitting. After training, the model was saved using Python's pickle module so it could be used later for prediction. The SVM **m**odel was selected instead of a CNN because it works better with smaller datasets, gives faster prediction results, and makes it easier to understand which image features are used for classification.

E. Blockchain Layer Design

Every prediction made by the model is stored as a new block in a locally hosted proof-of-work blockchain. Each block contains six pieces of information: a block number (Index), the date and time (Timestamp), the prediction details (disease name, confidence score, and image hash), the hash of the previous block, its own hash, and a nonce used to create a valid block. The blocks are saved in JSON format, which keeps the storage simple, lightweight, and easy to transfer between systems.

A new block is added to the blockchain only if its hash starts with the required number of leading zeros. Finding this hash requires some computation, which helps keep the blockchain secure. Since each block is connected to the previous one, changing any stored record will break the chain and make the change easy to detect. To hide such a change, someone would have to recreate that block and every block after it, which becomes more difficult as the blockchain grows. Creating a new block takes about 120–350 milliseconds, which fits well within the system's total processing time of 600–800 milliseconds.

F. System Integration and Workflow

The Flask backend provides two main services. The /predict endpoint receives a leaf image, identifies the disease, and returns the predicted disease name along with its confidence score. The /add block endpoint receives the prediction details and stores them as a new block in the blockchain after verification. The React frontend communicates with both endpoints using HTTP requests and displays the latest prediction and blockchain information on the user dashboard in real time. The complete working process of the system is shown in Algorithm 1.

Algorithm: AI-Blockchain Prediction Pipeline

```
 Input:  Leaf image uploaded via web interface
Output: Disease label, confidence score, blockchain block
1:  image ← capture_from_web_upload ()
2:  img_resized ← resize (image, 128×128)
3:  img_norm ← normalize(convert_colorspaces(img_resized))
4:  features ← extract_color_histograms(img_norm)  ‖
extract_texture(img_norm)
5:  features_scaled ← StandardScaler.transform(features)
6:  label, confidence ← svm_model.predict(features_scaled)
7:  block_data ← {label, confidence, timestamp(), hash(image)}
8:  blockchain.add_block(block_data)
9:  dashboard. display (label, confidence, blockchain. latest_hash())
```

G. Hardware and Software Environment

The system was developed and tested on a computer with an Intel Core i5 processor (2.5 GHz), 8 GB RAM, and Ubuntu 22.04 LTS operating system. The backend was built using Python 3.10 with Flask 3.0, Scikit-learn 1.4, OpenCV 4.8, NumPy 1.26, and Pandas 2.2. The frontend was developed using React.js 18 with the Material UI framework to create a simple and user-friendly interface. The blockchain data was stored in local JSON files. To test the system's performance in a different environment, it was also deployed on an AWS EC2 server with similar hardware specifications. The results were similar in both the local computer and the AWS server, showing that the system performed consistently.

© 2026 ADYPJIET, Volume 1(1), Online ISSN: 3049-1517, June 2026, pp. 38-49

A. Model Evaluation

The SVM model was tested using 160 test images and achieved an accuracy of 89.66%**.** The detailed results for each class are shown in Table II**.** The model performed best for Healthy leaves, with an F1-score of 0.92. It also identified Diabrotica speciosa leaf damage well, achieving an F1-score of 0.89**.** The Bacterial Blight class had a lower F1-score of 0.83 because it had fewer training images and the disease symptoms varied more from one leaf to another. The model achieved a weighted F1-score of 0.88**,** showing that it learned to identify different leaf conditions well instead of simply memorizing the training images. During five-fold cross-validation, the model achieved an average accuracy **of** 87.4% ± 1.8%**.** These consistent results indicate that the model performs reliably and is not significantly affected by overfitting.

TABLE II. Per- Class Classification Metrics

Algorithm	Accuracy (%)	Inference Time
k-Nearest Neighbors (k=3)	82.4	~180 ms
Logistic Regression	84.1	~60 ms
SVM with RBF Kernel (Ours)	**89.66**	**<100 ms**

B. Comparative Algorithm Analysis

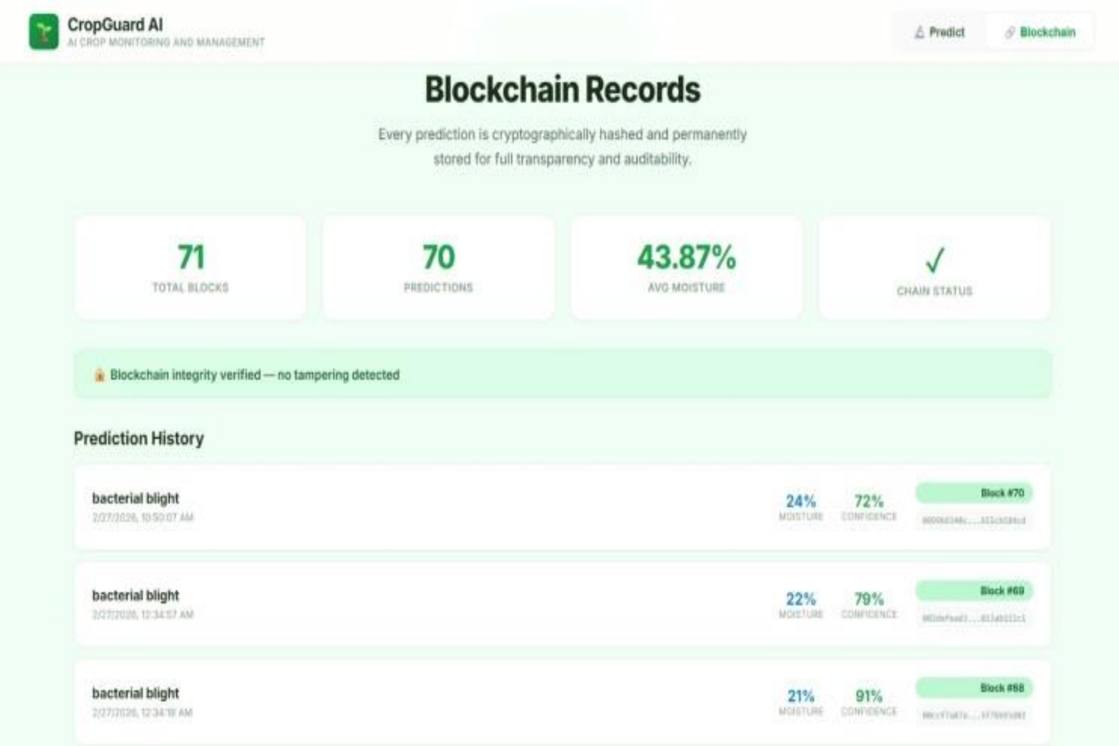
TABLE III. Algorithm Comparision

Class	Precision	Recall	F1-Score
Healthy	0.91	0.93	0.92
Bacterial Blight	0.85	0.82	0.83
Diabrotica Speciosa	0.90	0.88	0.89
Weighted Average	**0.88**	**0.87**	**0.88**

To compare the performance of the SVM model, two other machine learning models, k-Nearest Neighbors (k = 3) and Logistic Regression**,** were tested using the same dataset and features. The comparison results are shown in Table III**.** The SVM performed better than both models. It achieved 5.56% higher accuracy than k-NN and 7.26% higher accuracy than Logistic Regression**,** while still making predictions in less than 100 milliseconds. Although Convolutional Neural Networks (CNNs) can achieve very high accuracy (more than 99% in controlled environments), they require much larger datasets and powerful GPU hardware. These resources may not be available in the intended deployment environment. The SVM model was chosen because it provides a good balance of accuracy, fast prediction speed, and easy interpretation, making it a suitable choice for this prototype.

C. Blockchain Security Evaluation

A total of 100 predictions were stored in the blockchain one after another. Every new block was successfully added, and all blocks passed the integrity check. No errors or mismatches were found, showing that the blockchain stored the records securely and correctly. To test the security of the blockchain, one stored block was changed manually. As soon as the change was made, all the blocks after it became invalid because their hash values no longer matched. This showed that the blockchain could quickly detect any attempt to change the stored data.



This happens because the **SHA-256** algorithm generates a completely different hash value even if a single character in the data is changed. As a result, it is not possible to change a block without being detected. Any change immediately breaks the blockchain, making the tampering easy to identify.

D. System Latency and Resource Utilization

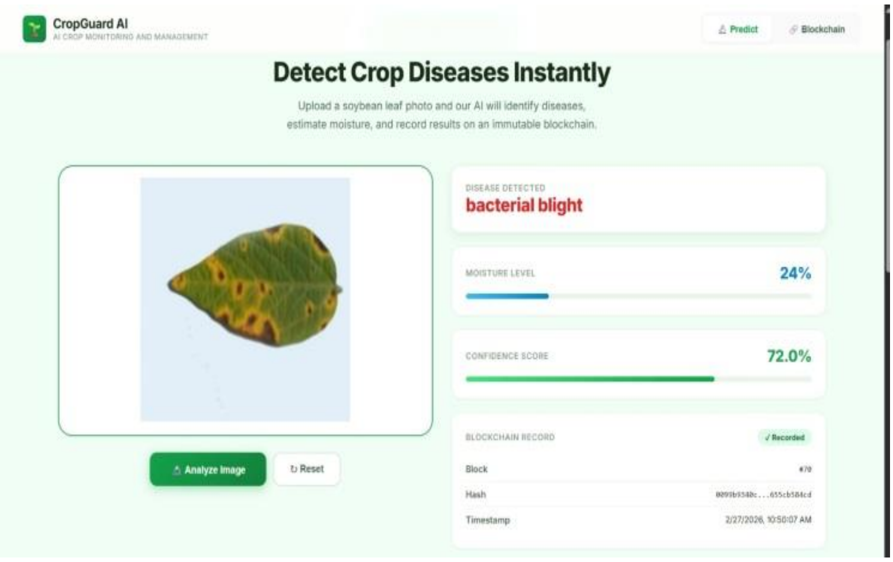
On average, the entire process takes 600–800 milliseconds, from uploading a leaf image to storing the result in the blockchain and updating the dashboard. Feature extraction and SVM prediction are

© 2026 ADYPJIET, Volume 1(1), Online ISSN: 3049-1517, June 2026, pp. 38-49

completed in less than 100 milliseconds. The remaining time is used for sending data between the frontend and backend, creating the blockchain block through the proof-of-work process, and updating the user interface. During testing, the CPU usage stayed below 40%**,** showing that the system did not put much load on the computer. Because the system uses very few resources, it can also run on low-cost devices such as a Raspberry Pi without a significant loss in performance.

E. Dashboard and User Interface

The React.js dashboard provides a simple and easy-to-use interface for farmers and other users. It displays the predicted disease, the confidence percentage, a color-coded health status, the block hash, the timestamp**,** and the blockchain integrity status**.** This allows users to quickly view the disease prediction and verify that the stored record is secure. Users can also view the blockchain details for any prediction directly from the dashboard. This makes it easy to check and verify the stored records without needing any technical knowledge about blockchain or cryptography. The system works through a web browser, so users do not need to install any software. It can be used on any device with a camera, including smartphones. This makes the system easy to access and suitable for farmers in rural areas.



5.0 Conclusion and Future Work

This paper presented and tested a working prototype that combines AI-based plant disease detection with blockchain technology in a single system. The SVM model successfully identified soybean leaf diseases with an accuracy of 89.66%**,** while the blockchain securely stored every prediction without any loss of data integrity. A simple React.js dashboard allows farmers to easily upload leaf images, view disease predictions, and verify the stored records. The results show that an accurate and easy-to-understand machine learning model, when combined with blockchain, can provide secure and reliable disease detection. This approach has the potential to support precision agriculture by helping farmers make better decisions with trustworthy data.

This work produced four main outcomes. First, it developed a complete system that combines AI-based disease detection with blockchain storage and can be used without any special hardware. Second, it introduced a 296-feature extraction method that performs well even when only a small amount of labelled training data is available, making it a practical alternative to more complex deep learning models. Third, the study developed a proof of work blockchain that was tested and successfully detected any attempt to change the stored data. Fourth, it created a web-based dashboard that allows farmers and other users to easily view the disease prediction and verify that the stored record is secure and has not been modified.

There are several ways to improve this work in the future. One option is to replace the SVM with a CNN or Vision Transformer model and train it on a larger dataset, such as Plant Village, to improve disease detection accuracy. Another improvement is to add IoT sensors that measure temperature, humidity, and soil pH. This additional environmental information can help the system make more accurate predictions because leaf images alone may not provide all the necessary details. In the future, the system can be run on low-cost edge devices such as a Raspberry Pi or NVIDIA Jetson**.** This will reduce the need for cloud access and make the system more useful in areas with poor internet connectivity. The local blockchain can also be replaced with Hyperledger Fabric**,** allowing multiple users or organizations to securely share and verify the same records instead of maintaining separate copies. A mobile application for Android or iOS would be more convenient for farmers than using a web browser in the field. The system should also be tested on different crops in real farming conditions with the help of agricultural institutions to evaluate its performance outside the laboratory. In addition, using IPFS (Inter Planetary File System) to store images would provide a better way to manage high-resolution leaf images instead of storing them only on local computers.

References:

1. Assimakopoulos and C. Vassilakis, AI and Related Technologies in the Fields of Smart Agriculture: A Review, Information, vol. 16, no. 2, p. 100, 2025. https://doi.org/10.3390/info16020100.

2. J. Botero-Valencia et al., Machine Learning in Sustainable Agriculture: Systematic Review and Research Perspectives, Agriculture, vol. 15, no. 4, p. 377, 2025. https://doi.org/10.3390/agriculture15040377.

3. T. Miller et al., The IoT and AI in Agriculture: The Time Is Now—A Systematic Review of Smart Sensing Technologies, Sensors, vol. 25, no. 12, p. 3583, 2025. https://doi.org/10.3390/s25123583.

4. K. Demestichas and N. Peppes, Blockchain Technology and Artificial Intelligence Together: A Critical Review on Applications, Applied Sciences, vol. 12, no. 24, p. 12948, 2022. https://doi.org/10.3390/app122412948.

5. Y. Yang, M. Lin, Y. Lin, C. Zhang, and C. Wu, A Survey of Blockchain Applications for Management in Agriculture and Livestock Internet of Things, Future Internet, vol. 17, no. 1, p. 40, 2025. https://doi.org/10.3390/fi17010040.

6. A. Gutub, Leveraging Deep Learning for Real-Time Coffee Leaf Disease Identification, Agriculture, vol. 14, no. 3, p. 456, 2024. https://doi.org/10.3390/agriculture14030456.

7. R. M. Ellahi, L. C. Wood, and A. E.-D. A. Bekhit, Blockchain-Driven Food Supply Chains: A Systematic Review for Unexplored Opportunities, Applied Sciences, vol. 14, no. 19, p. 8944, 2024. https://doi.org/10.3390/app14198944.

8. K. Demestichas, N. Peppes, T. Alexakis, and E. Adamopoulou, Blockchain in Agriculture Traceability Systems: A Review, Applied Sciences, vol. 10, no. 12, p. 4113, 2020. https://doi.org/10.3390/app10124113.

9. M. S. Hossain et al., IDairy: Intelligence and Secure E-Commerce Platform for Dairy Products Using Blockchain Technology, IEEE Access, vol. 10, pp. 57050–57067, 2022. https://doi.org/10.1109/ACCESS.2022.3174112.

10. L. B. Krithika, Survey on the Applications of Blockchain in Agriculture, Agriculture, vol. 12, no. 9, p. 1333, 2022. https://doi.org/10.3390/agriculture12091333.

11. J. Kim and S. Lee, Smart Packaging with Blockchain: Integrating Sensors for Traceability, IEEE Transactions on Industrial Informatics, vol. 19, no. 5, pp. 5678–5689, 2023. https://doi.org/10.1109/TII.2023.3245678.

12. K. Salah, Blockchain in Agriculture, in Blockchain Technologies for Sustainability, Springer, 2020, pp. 55–78.

13. A. Iftikhar and M. Khan, Raspberry Pi Smart Sensor Node Design for Perishable Food Monitoring, IEEE Sensors Journal, vol. 20, no. 15, pp. 8765–8774, 2020. https://doi.org/10.1109/JSEN.2020.2987654.

14. F. Pedregosa et al., Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

15. G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools, 2000.