International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 11, Issue 2, 2025, PP 1-11
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
DOI: https://doi.org/10.20431/2349-4859.1102001
www.arcjournals.org

A Cloud-Based Collaborative Platform with Artificial Intelligence for Farmers to Identify Track and Forecast Plant Diseases

Gaddipati Navya Kranthi, G Bharathi Vunnava, Dinesh Babu

Department of CSE, R V Institute of Technology, Guntur

Corresponding Author:** ***Gaddipati Navya Kranthi, *Department of CSE, R V Institute of Technology, Guntur*

Abstract: *Plant diseases pose a serious risk to the environment, farmers, consumers, and the world economy. In India alone, pests and diseases destroy 35% of field crops, costing farmers money. Since many pesticides are poisonous and biomagnified, their indiscriminate use poses a major health risk as well. Targeted therapies, crop surveillance, and early disease diagnosis can prevent these negative effects. Experts in agriculture diagnose the majority of diseases by looking at outward signs. Experts are hard to come by for farms, though.*

For automated disease detection, tracking, and forecasting, our initiative is the first collaborative, integrated platform. By taking pictures of the afflicted plant portions, farmers can use a mobile app to quickly and precisely identify diseases and obtain remedies. The most recent Artificial Intelligence (AI) techniques for cloud-based image processing make real-time diagnosis possible. To improve its accuracy, the AI model is constantly learning from user-uploaded photos and professional recommendations. The app also allows farmers to communicate with local specialists. A cloud-based repository of geo-tagged photos and micro-climatic variables is used to create disease density maps with spread forecasts for preventative purposes.

Experts can do disease analytics using geographic representations using a web interface. Large disease datasets were generated using plant photos that were self-collected from numerous farms over a period of seven months in order to train the AI model (CNN) in our research. The automatic CNN model was used to diagnose test photos, and plant pathologists verified the findings. The accuracy of illness identification was over 95%. Our system can be implemented as a cloud-based service for farmers and professionals for environmentally sustainable crop production. It is a new, scalable, and easily accessible tool for managing diseases of various agricultural crop plants.

Keywords: *Plant diseases, AI model, micro-climatic variables*

1. Introduction

Agriculture plays a vital role in sustaining human life, particularly in densely populated developing nations like India, where improving the yield and quality of crops, fruits, and vegetables is crucial. However, these two objectives—higher productivity and better quality—are often hindered by the occurrence of plant diseases. Many of these diseases are infectious in nature and, if not detected early, can result in complete crop failure.

Traditional methods of disease detection that rely on human experts face significant limitations. The sheer geographical spread of farmland, coupled with the limited number of trained plant pathologists and low levels of agricultural education among farmers, makes human-based diagnosis slow, costly, and inefficient. There is, therefore, an urgent need to introduce automation into crop disease detection using modern technology.

Recent years have seen advances in agricultural technology through the use of robotics, computer vision, and other smart systems. While these technologies have been successfully applied to monitor plant growth, manage nutrients, and detect weeds, the automation of plant disease identification is still underdeveloped. This is despite the fact that many plant diseases can be visually diagnosed based on changes in color, shape, and other physical symptoms.

One major reason for the slow adoption of such technologies in agriculture is the lack of large-scale commercial investment, especially when compared to fields like healthcare and education. Moreover, difficulties in connecting farmers with specialists and the high cost of deploying advanced solutions have limited progress.

However, the rise of mobile phones, cloud computing, and artificial intelligence (AI) presents a unique opportunity to overcome these challenges. In countries like India, low-cost smartphones with internet access, cameras, and GPS are now widely available. Farmers can use these devices to capture images of infected crops and upload them—along with their location data—to cloud-based platforms.

These platforms, powered by high-performance computing, can process images, maintain centralized databases, and perform complex data analytics. AI, particularly deep learning models like Convolutional Neural Networks (CNNs), has proven highly effective at image classification tasks—often surpassing human performance. CNNs mimic the structure of the human brain and are trained on large datasets to recognize patterns in new images. Since the success of "AlexNet" in the 2012 ImageNet competition, CNNs have become the preferred model for image-related AI applications.

The evolution of AI has been fueled by better computational resources, more extensive labeled datasets, and improved algorithms. Furthermore, open-source frameworks like TensorFlow have made it easier and more affordable to build and deploy AI systems.

Previous research efforts have explored the collection of plant images, texture analysis, RGB imaging, spectral pattern recognition, and fluorescence-based diagnostics. While neural networks have been applied in the past, they typically focused on basic texture recognition. Our approach goes further by integrating modern mobile technology, cloud services, and advanced AI to deliver a complete, automated disease diagnosis system. This system replicates the diagnostic capabilities of plant pathologists and makes them directly accessible to farmers. It also supports collaboration, allowing ongoing data collection and expert input to improve the model's accuracy and help monitor disease outbreaks effectively.

2. Proposed Method

In this project, the author utilizes Convolutional Neural Networks (CNNs), a branch of Artificial Intelligence, to train on a dataset of plant disease images. Once trained, the CNN model can accurately predict the type of disease present in new images uploaded by users. To store both the trained CNN model and image data, cloud services are employed, ensuring scalable and secure data management.

Although the initial idea involves using smartphones to capture and upload images, developing a dedicated Android application would require additional time and resources. To overcome this, the project is implemented as a Python-based web application. This web app allows users to upload images of plants, after which the trained CNN model processes the image and identifies any present diseases.

If deployed on a live web server, the application can also extract the user's location from the HTTP request metadata. This geolocation feature can be used to map disease occurrences, helping visualize affected areas geographically.

3. Feasibility Study

During this phase, the project's feasibility is evaluated, and a preliminary business proposal is presented, including a general implementation plan and rough cost estimates. The purpose of this analysis is to determine whether the proposed system is practical and cost-effective, ensuring it does not become a burden on the organization. A basic understanding of the system's primary requirements is necessary to conduct a meaningful feasibility study.

The three key considerations in feasibility analysis are

3.1. Economical Feasibility

3.2. Technical Feasibility

3.3. Social Feasibility

A feasibility study assesses whether a proposed system can be developed and implemented successfully within the constraints of the organization. This includes evaluating the economic, technical, and social aspects to ensure the system is practical, sustainable, and accepted by users.

3.1 Economic Feasibility

This analysis evaluates the financial impact of the system on the organization. Given the limited budget available for research and development, it is crucial that all expenditures are justified. In this project, cost-efficiency was achieved by leveraging open-source and freely available technologies. Only a few customized components required purchase, ensuring that the overall development remained well within budget.

3.2 Technical Feasibility

The technical feasibility study examines whether the system’s technical requirements align with the existing infrastructure and resources. The system should not place excessive demands on current technical capabilities. The proposed solution has minimal hardware and software prerequisites, making it easy to integrate without requiring significant modifications or upgrades. This ensures smooth implementation with little to no disruption to existing operations.

3.3 Social Feasibility

This aspect of the study evaluates the system's acceptance among its intended users. It is important that users feel comfortable with the system and view it as a helpful tool rather than a threat. Effective training and user education are key to building confidence and encouraging constructive feedback. A system that is well understood by its users is more likely to be adopted and successfully utilized.

4. SYSTEM TESTING

Testing is a critical phase in software development, aimed at identifying and resolving errors before deployment. The primary goal is to ensure the system meets its specified requirements and performs reliably under expected conditions. Testing involves verifying both individual components and the integrated system to detect any functional weaknesses or failures.

There are various types of testing, each tailored to specific objectives, including:

- **Unit Testing:** Verifies the functionality of individual modules or components.
- **Integration Testing:** Ensures that combined modules function together as expected.
- **System Testing:** Validates the complete and integrated software system against the defined requirements.
- **User Acceptance Testing (UAT):** Confirms the system meets user needs and expectations in a real-world environment.

5. TYPES OF TESTING

Effective testing is essential to ensure software quality and system reliability. The following are key testing types applied throughout the development life cycle:

5.1. Unit Testing

Unit testing focuses on validating individual components or units of the software. It ensures that internal logic, decision branches, and inputs produce the expected outputs. This is typically a white-box testing method performed after each module is developed and before integration. Unit tests are designed based on the software's structure and logic, ensuring each business rule or function operates as expected.

- **Objective:** Verify that each component performs its intended function correctly.
- **Scope:** Isolated units/modules.
- **Approach:** Structural and invasive.

5.2. Integration Testing

Integration testing evaluates the interaction between multiple software modules after individual units have passed unit testing. It ensures that combined components work together as intended, highlighting issues arising from module integration.

- **Objective:** Validate data flow and interaction between integrated units.

- **Scope:** Multiple modules working as a complete subsystem.
- **Approach:** Interface-level, event-driven.

5.3. Functional Testing

Functional testing verifies that the system behaves according to defined business and technical requirements. It focuses on validating inputs, outputs, and core system functions based on user expectations and use cases.

Key Aspects Covered:

- **Valid Inputs:** Accepted and processed correctly.
- **Invalid Inputs:** Rejected appropriately.
- **Functionality:** All identified functions are executed correctly.
- **Output Validation:** Output results match expected values.
- **Interfacing:** Connected systems or APIs are triggered and responded to as required.

5.4. System Testing

System testing assesses the complete and integrated system to ensure it meets the specified requirements. This testing phase emphasizes verifying end-to-end system functionality and ensures known inputs produce expected results.

- **Objective:** Validate the entire software system.
- **Approach:** Process and configuration-oriented.

5.5. White Box Testing

White box testing is a method where the tester has knowledge of the software's internal logic, structure, and code. It allows the tester to design cases that explore the code thoroughly, ensuring internal paths function as expected.

- **Approach:** Code-aware; tests include logic branches, loops, and internal operations.
- **Used For:** Developers testing logic during development.

5.6. Black Box Testing

In black box testing, the tester has no internal knowledge of the code structure. Tests are performed from the user’s perspective, focusing solely on inputs and expected outputs.

- **Approach:** Interface-based; internal code is treated as a "black box."
- **Used For:** Functional testing, user acceptance testing.

5.6. Test Strategy and Approach

- Manual field testing is performed to simulate real-world user interactions.
- Functional tests are written in detail for each feature.

5.5. Test Objectives

- All input fields should accept valid entries.
- Each link should correctly open the corresponding page.
- User interaction should be smooth with no noticeable delays in responses.

5.6. Features to Be Tested

- Input validation (format and type).
- Duplicate entry prevention.
- Correct page navigation via links.

5.7. Integration Testing

Integration testing is conducted incrementally to ensure combined software modules function without interface-related issues. This process identifies any errors in the interaction between components.

- **Objective:** Verify proper communication between integrated software components.
- **Result:** All integration test cases executed successfully with no defects found.

5.8. Acceptance Testing

User Acceptance Testing (UAT) is the final validation performed by end users to confirm the system meets all functional requirements and is ready for deployment.

- **Objective:** Validate the software against user expectations and business needs.
- **Result:** All UAT test cases passed successfully; no defects encountered.

6. Conclusion

This paper introduces a fully automated, cost-effective, and user-friendly end-to-end system aimed at addressing a major issue faced by farmers in agriculture: the need for accurate, rapid, and early detection of crop diseases, along with awareness of potential disease outbreaks. Such timely identification can significantly aid in making informed decisions for effective disease management.

The proposed solution enhances existing research by integrating deep Convolutional Neural Networks (CNNs) for disease identification, incorporating a collaborative platform to improve model accuracy over time, and utilizing geo-tagged images to generate disease distribution maps. Additionally, the system features an expert interface to support analytical insights.

A high-performance deep CNN model, specifically the "Inception" architecture, is used for real-time image classification. This classification is performed on a cloud-based backend, while the front-end is accessible through a mobile-friendly interface. The collaborative component allows users to contribute images, which are then used to continuously expand the training dataset, thereby enhancing the model's precision through periodic retraining.

Furthermore, the inclusion of geolocation data in uploaded images facilitates the generation of disease density visualizations, helping to identify regional outbreaks. The proposed approach proves to be scalable due to its cloud infrastructure, and its CNN model maintains high accuracy across diverse disease types. Experimental results also confirm its ability to recognize early symptoms and differentiate between similar diseases within the same family, making it a viable and practical solution for real-world agricultural challenges.

References

[1] L. Saxena and L. Armstrong, "A survey of image processing techniques for agriculture," in Proceedings of Asian Federation for Information Technology in Agriculture, 2014, pp. 401-413.

[2] E. L. Stewart and B. A. McDonald, "Measuring quantitative virulence in the wheat pathogen Zymoseptoriatritici using high-throughput automated image analysis," in Phytopathology 104 9, 2014, pp. 985– 992.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems, 2012.

[4] TensorFlow.[Online].Available:https://www.tensorflow.org/

[5] D. P. Hughes and and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," in CoRR abs/1511.08060, 2015.

[6] S. Raza, G. Prince, J. P. Clarkson and N. M. Rajpoot, "Automatic detection of diseased tomato plants using thermal and stereo visible light images," in PLoS ONE, 2015.

[7] D. L. Hernández-Rabadán, F. Ramos-Quintana and J. Guerrero Juk, "Integrating soms and a bayesian classifier for segmenting diseased plants in uncontrolled environments," 2014, in the Scientific World Journal, 2014.

[8] S. Sankaran, A. Mishra, J. M. Maja and R. Ehsani, "Visible-near infrared spectroscopy for detection of huanglongbing in citrus orchards," in Computers and Electronics in. Agriculture 77, 2011, pp. 127–134.

[9] C. B. Wetterich, R. Kumar, S. Sankaran, J. B. Junior, R. Ehsani and L. G. Marcassa, “A comparative study on application of computer vision and fluorescence imaging spectroscopy for detection of huanglongbing citrus disease in the USA and Brazil,” in Journal of Spectroscopy, 2013.

[10] C. Szegedy,“Rethinking the inception architecture for computer vision,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818-2826.

[11] Mango Diseases and Symptoms. [Online].Available: http://vikaspedia.in/agriculture/crop-production/integrated-pestmanagment/ipm-for-fruit-crops/ipm-strategies-for-mango/mangodiseases-and-symptoms

[12] P. Subrahmanyam, S. Wongkaew, D. V. R. Reddy, J. W. Demski, D. McDonald, S. B. Sharma and D. H. Smith, "Field Diagnosis of Groundnut Diseases”. Monograph. International Crops Research Institute for the Semi-Arid Tropics, 1992.

Citation: *Gaddipati Navya Kranthi et., al (2025). “A Cloud-Based Collaborative Platform with Artificial Intelligence for Farmers to Identify Track and Forecast Plant Diseases”. International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), vol 11, no. 2, 2025, pp. 1-11. DOI: https://doi.org/10.20431/2349-4859.1102001.*