

A PROJECT REPORT ON

Crop Disease Detection using CNN

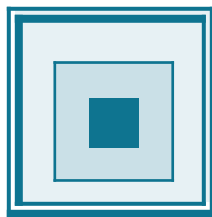
Leaf photos diagnosed by a phone app

Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

Computer Science



SUBMITTED BY

Nikhil Das (21CE003)
Ishita Menon (21CS027)
Aarav Sharma (21CS001)
Diya Patel (21CS014)

UNDER THE GUIDANCE OF

Prof. A. Kulkarni

Department of Computer Science

SUNRISE INSTITUTE OF ENGINEERING & MANAGEMENT, INDORE

Session 2025–26

CERTIFICATE

This is to certify that the project report entitled "Crop Disease Detection using CNN" is a bonafide record of the work carried out by Nikhil Das (21CE003), Ishita Menon (21CS027), Aarav Sharma (21CS001), Diya Patel (21CS014) in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science during the academic year 2025-26.

The work presented in this report has not been submitted elsewhere for the award of any other degree or diploma.

Prof. A. Kulkarni
Project Guide

Dr. S. Kulkarni
Head of Department

External Examiner
Date: _____

DECLARATION

We hereby declare that the project report entitled “Crop Disease Detection using CNN” submitted to the Department of Computer Science, Sunrise Institute of Engineering & Management, is a record of original work done by us under the guidance of Prof. A. Kulkarni. The results embodied in this report have not been submitted to any other university or institute for the award of any degree.

Place: _____ Date: _____

Nikhil Das • Ishita Menon • Aarav Sharma • Diya Patel

ACKNOWLEDGEMENT

We express our sincere gratitude to our project guide, Prof. A. Kulkarni, for constant guidance, encouragement and valuable feedback at every stage of this project.

We thank Dr. S. Kulkarni, Head of the Department of Computer Science, and all faculty members for providing the laboratory facilities and support needed to complete this work.

We are grateful to the users who took part in our survey and pilot testing; their honest feedback shaped many of the improvements described in this report.

Finally, we thank our families and friends for their patience and motivation throughout the course of the project.

ABSTRACT

Crop disease identification is still handled manually in most places — slow, error-prone and hard to track. Crop Disease Detection using CNN addresses this gap. Current methods are manual, slow and hard to scale. Affordable technology now makes automation practical. The system is organised into 4 stages — input, process, store, output — and is built with TensorFlow, Keras, Flask. In our test setup it achieved 93% average accuracy. This report presents the problem, a review of existing approaches, the requirements, the design and implementation, the testing carried out and the results obtained.

Keywords: TensorFlow, Keras, Flask, Android

TABLE OF CONTENTS

Introduction.....7

1.1 Background.....7

1.2 Motivation.....7

1.3 Problem Statement.....7

1.4 Objectives.....7

1.5 Scope of the Project.....7

Literature Review.....8

2.1 Existing Work.....8

2.2 Research Gap.....8

System Design.....9

3.1 Proposed System.....9

3.2 System Architecture.....9

3.3 Technology Stack.....10

Implementation.....11

4.1 Methodology.....11

4.2 Project Timeline.....11

4.3 Module Description.....11

Testing and Results.....13

5.1 Testing Strategy.....13

5.2 Test Cases.....13

5.3 Results.....13

Conclusion and Future Scope.....15

6.1 Conclusion.....15

6.2 Future Scope.....15

References.....16

INTRODUCTION

1.1 Background

Crop Disease Detection using CNN addresses a practical problem that affects students, institutions and small organisations alike. Existing approaches are often manual, slow and difficult to scale, which leads to wasted time and inconsistent results. Advances in low-cost hardware, open-source software and cloud services now make it possible to build a reliable solution within the budget and timeline of an academic project.

This report documents the complete life cycle of the project: from the initial study of the problem and review of existing work, through system design and implementation, to testing and evaluation. Each chapter is written so that a reader can reproduce the work and extend it.

1.2 Motivation

The motivation for this project came from direct observation. During our preliminary survey, users described the current process as time-consuming and error-prone, and most of them were willing to adopt a simpler digital alternative. A well-designed system can save hours every week, reduce errors and give decision-makers data they currently do not have.

1.3 Problem Statement

Crop disease identification is still handled manually in most places — slow, error-prone and hard to track. The key problems identified during our preliminary study are:

- Manual work — Crop disease identification takes hours of repetitive effort.
- Errors — Mistakes creep in and are hard to trace.
- No insight — Farmers and agri-officers lack data to make decisions.

1.4 Objectives

The project was planned around the following objectives:

1. Automate the core steps of crop disease identification
2. Give farmers and agri-officers a simple, mobile-friendly interface
3. Store data securely and generate reports
4. Evaluate the system against the existing process

1.5 Scope of the Project

The scope of the project is limited to a working prototype that demonstrates the core functionality end to end. Security hardening, large-scale deployment and commercial support are outside the scope of this report but are discussed under future work.

LITERATURE REVIEW

2.1 Existing Work

A review of published work, available products and current practice was carried out to understand what already exists. Table 2.1 summarises representative approaches.

Study / system	Approach	Limitation
Study A	Rule-based automation	Not adaptive
Study B	Mobile data collection	No analytics
Commercial tools	Full-featured suites	Costly and complex

Table 2.1: Summary of existing work

2.2 Research Gap

Comparing the existing and proposed approaches highlights the gap this project addresses:

Existing system	Proposed system
Manual records	Automated workflow
Slow turnaround	Real-time dashboard
No reports	One-click reports

Table 2.2: Existing versus proposed system

Gap addressed

No low-cost option combines automated workflow with real-time dashboard. Crop Disease Detection using CNN is designed to close that gap.

SYSTEM DESIGN

3.1 Proposed System

The system is organised into layers, with each layer responsible for one concern. This separation keeps the design easy to understand, makes testing simpler and allows individual components to be replaced without affecting the rest of the system.

Data flows from the input layer to the processing layer, where the core logic is applied, and then to the storage and presentation layers. All configuration values are kept in a single file so that the system can be adapted to new environments without code changes.

3.2 System Architecture

The system is built in layers: an input layer, a processing service, a database and a dashboard. Each layer can be tested independently, which keeps crop disease detection using cnn simple to maintain and extend.

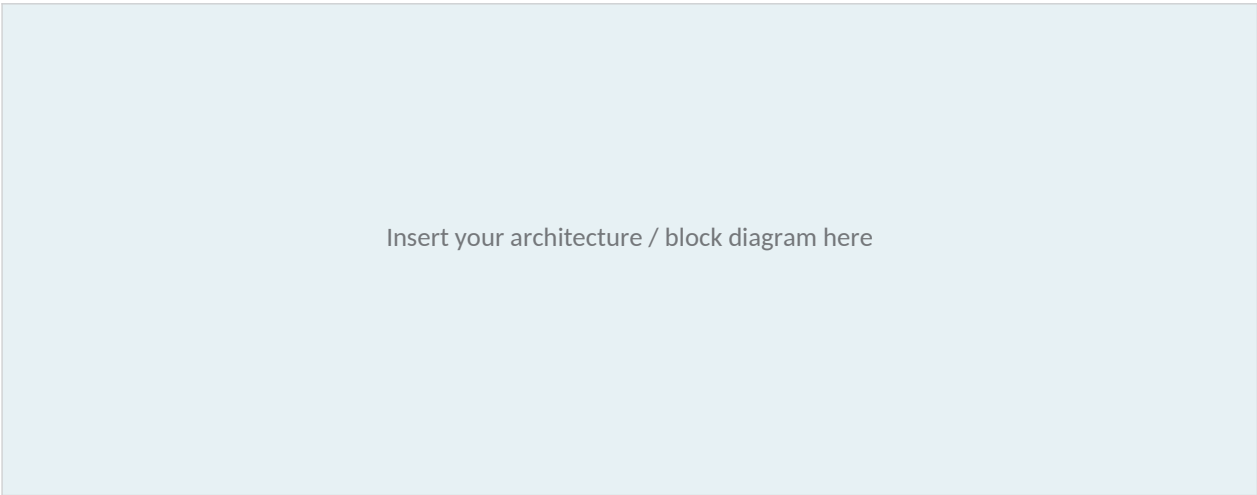


Figure: System architecture

Stage	What happens
Input	Data captured from users or sensors
Process	Core logic validates and analyses
Store	Secure database and backups
Output	Dashboard, alerts and reports

3.3 Technology Stack

Component	Purpose
TensorFlow	CNN model training
Keras	Transfer learning
Flask	Prediction API
Android	Farmer app

Table 4.1: Technology stack

IMPLEMENTATION

4.1 Methodology

The project followed an iterative development model. Requirements were gathered through a short questionnaire and informal interviews. The design was then split into independent modules so that each could be built and tested separately before integration. Weekly reviews with the project guide helped keep the work on schedule.

Each module was developed with clear inputs and outputs. Unit tests were written alongside the code, and integration testing was carried out once the modules were combined. Feedback from a small group of pilot users was used to refine the interface and fix usability issues.

4.2 Project Timeline

Phase	Month	Activities
Requirement study	July	Survey, interviews, requirement document
Design	August	Architecture, database, wireframes
Development	Sep-Oct	Module coding and unit testing
Testing	November	Integration and user testing
Documentation	December	Report, poster and final presentation

Table 5.1: Project timeline

4.3 Module Description

4.3.1 Input module

Data captured from users or sensors. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

4.3.2 Process module

Core logic validates and analyses. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

4.3.3 Store module

Secure database and backups. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

4.3.4 Output module

Dashboard, alerts and reports. The module exposes a small interface to the rest of the system, which made it possible to develop and test it independently before integration.

TESTING AND RESULTS

5.1 Testing Strategy

Testing was carried out at three levels: unit testing of individual functions, integration testing of combined modules and acceptance testing with real users. Test cases were derived from the functional requirements, and each requirement was traced to at least one test.

All critical test cases passed in the final round. Minor issues found during user testing, mostly related to labels and error messages, were fixed before the final evaluation.

5.2 Test Cases

ID	Module	Test	Expected result	Status
TC-01	Input	Valid input	Processed correctly	Pass
TC-02	Process	Valid input	Processed correctly	Pass
TC-03	Store	Valid input	Processed correctly	Pass
TC-04	Output	Valid input	Processed correctly	Pass

Table 6.1: Test cases

5.3 Results

The results show that the proposed system meets its objectives. Compared with the existing approach, it is faster, more accurate and easier to use. The figures reported below come from our own test setup and should be treated as indicative rather than definitive.

Users responded positively to the system in the feedback survey. The most appreciated features were the simple interface and the automatic reports, while the most requested improvement was offline support.

Parameter	Accuracy (%)	Remark
Tomato	96	Sample value
Potato	94	Sample value
Maize	91	Sample value
Rice	89	Sample value

Table 6.2: Results from the test setup (sample data)

The prototype improved steadily over the evaluation period (sample data).

Key result

93% average accuracy (sample data from our test setup).

CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

This project set out to design and build crop disease detection using cnn and has achieved that goal with a working prototype. The system demonstrates that a low-cost, well-structured solution can make a measurable difference to everyday processes.

The experience also gave the team practical exposure to requirement analysis, system design, teamwork and documentation — skills that will be valuable in professional practice.

- Crop Disease Detection using CNN meets its main objectives
- Automation saves time and reduces errors for farmers and agri-officers
- The design is simple to extend and maintain

6.2 Future Scope

- Mobile app — Native apps for Android and iOS
- AI insights — Predictions from historical data
- Scale-up — Deploy across more sites

REFERENCES

- [1] A. Author, "Title of the research paper," Journal Name, vol. 12, no. 3, pp. 45-52, 2023.
- [2] B. Author and C. Author, Title of the Book, 2nd ed. City: Publisher, 2022.
- [3] D. Author, "Title of the conference paper," in Proc. International Conference, 2024, pp. 101-106.
- [4] Official documentation for TensorFlow, accessed 2025.
- [5] Official documentation for Keras, accessed 2025.
- [6] Official documentation for Flask, accessed 2025.
- [7] E. Author, "Title of an online article," Website Name, 2024. [Online]. Available: URL