AI Plant Disease Detection App
Vision Document

Version 1.0

Team Members:

Joshua Hanson
Mason Lewis
David Markowski-Goldwyn
Eh Doh Kue Kyi
Matthew Murphy

Title: Vision Document	
Version: 1.0	Date: 10/9/25

Revision History

Date	Version	Description	Name
10/5/25	1.0	Rough Draft	Mason Lewis

Table of Contents

Introduction

1.1 Overview

The AI plant disease detection app is a prototype using image recognition and machine learning to help farmers, gardeners, and researchers quickly identify plant diseases. Users will take or upload a photo of the affected plant, and the app analyzes the visible symptoms. Users will then submit any information about the plant that cannot be analyzed through the

photo to increase the confidence of the prediction. It will then provide a diagnosis, along with treatments or preventive measures for the disease.

1.2 Definitions, Acronyms, Abbreviations

- AI/Artificial Intelligence - technology that enables machines to perform tasks that normally require human intelligence.
- ML/Machine Learning - a subset of AI that enables systems to learn and improve from data
- CNN/Convolutional Neural Network – a deep learning algorithm used for analyzing images
- OpenCV – a programming library used for real-time image processing
- TensorFlow – An open-source machine learning framework used for building and training AI models
- Experta – A Python expert system framework used to represent knowledge-based reasoning
- UI/User Interface – The part of the app that users interact with, including buttons, menus, and visual elements
- Cloud Service Provider – A company or platform that offers cloud computing for hosting, storage, and data processing
- Dataset – A structured collection of related data used to train and evaluate an AI model

● Problem Statement

The problem of	● Difficulty in accurately diagnosing plant disease ● Limited access to agricultural experts ● Misidentification can lead to inappropriate use of pesticides
Affects	● Farmers

	• Gardeners • Environment
The impact of which is	• Spread of plant diseases • Reduced yield of crops • Environmental pollution • Reduced food quality
A successful solution would be	• An AI-based app that can quickly and accurately provide diagnoses and recommendations.

• Stakeholder and User Descriptions

3.1 Stakeholder summary

Name	Description	Responsibility
Dr. Hajiarbabi	PFW Project Advisor	Assist the team in completing the project, provide the necessary resources for developing the application, and monitor progress.

3.2 User Environment

The user begins on the main page, where they can choose to start a new diagnosis or view a previous one. Selecting a previous diagnosis will show a list of all diagnoses performed on the device. Starting a new diagnosis allows the user to take or upload pictures; multiple pictures can be uploaded at once. After providing one or more pictures, the system will diagnose a disease and display its symptoms and treatments.

3.3 Operating Environment

- Cloud Service Provider: PFW
- ML Training: TensorFlow/Keras

- Expert System: Experta
- Image Processing: OpenCV
- Mobile Developer: React Native/Android Studio

3.4 Key Stakeholder or User Needs

Need	Priority	Concerns	Proposed Solution
Demonstrate a prototype system	Critical	Ensuring that the prototype accurately detects diseases	Develop a functional prototype using machine learning
Accurate and fast disease diagnosis	Critical	Misidentification could lead to crop loss	Use an AI model trained on large, verified datasets
Easy-to-use interface	Critical	Users may have limited technical skills	Design an intuitive interface with clear instructions
Offline Accessibility	Medium	Many users will operate in areas with a poor internet connection	Run the CNN locally on the phone using MobileNet

Product Overview

4.1 Product Features

Critical Features

- Take picture
- Upload one or more pictures
- Input data that cannot be gathered from images
- Display diagnoses with treatments
- Diagnose with 85% or greater accuracy
- Store diagnoses to revisit

Optional Features

- Chatbot integration
- Prediction using historical data
- Prevention plans

- Language support

4.2 Assumptions and dependencies

Assumptions

- Farmers and gardeners are assumed to have access to smartphones or tablets capable of running the app.
- While offline use may be supported, it's assumed that users can occasionally connect to the internet

Dependencies

- An Android device with internet access
- Git repository for code management
- TensorFlow for machine learning

Title: Vision Document	
Version: 1.0	Date: 10/9/25

Intelligent Plant Disease

Project Glossary

Team Members:

Josh Hanson

Eh Doh Kue Kyi

Mason Lewis

David Markowski

Matthew Murphy

Glossary

A. **TensorFlow** is an open-source machine learning library with multilanguage support, optimization for NVIDIA GPUs, and interoperability with the NumPy library

B. **NumPy (Numerical Python)** is a mathematics library for Python commonly used for arrays and matrices

C. **Scikit-learn** is a machine learning Python library commonly used for statistical modeling

D. **Matplotlib** is a Python library commonly used to plot and display graphs and functions visually

E. **Pandas** is a Python library built on top of NumPy that is used to process data and display data visually via DataFrame

F. **Git** is a command line based development control system in which users can track changes to and modify their source code

G. **GitHub** is a visual Git representation with a user interface and common UI elements

H. **Hosting** refers to storing files on a server to view, manipulate, and retrieve

I. **Cloud** refers to hosting files and data on a non-local, remote, or external server which is not owned or managed by the people or organizations hosting the data

J. **ComputerVision** is a field of artificial intelligence with a focus on image manipulation and identification

K. **OpenCV** is an open-source C++ ComputerVision library that is compatible with multiple languages

L. **React Native** is a mobile application development framework used to create applications for iOS and Android

M. **Convolutional Neural Networks (CNNs)** are deep learning models which use layering to recognize patterns

N. **Recurrent Neural Networks (RNNs)** are artificial neural networks which can remember contextual information

O. **Expert Systems** are decision making programs which use facts and rules to come to a specific conclusion, mimicking a human expert

P. **User Interface** refers to the graphical, non-command line based interface where users can interact with an application without modifying code or running specific commands

Q. **Knowledge Bases** contain facts and rules which are processed by an inference engine in an expert system

R. **Inference Engines** combine the facts and rules contained within the knowledge base with reasoning algorithms to allow the expert system to come to a specific conclusion

S. **Hybrid Systems** combine expert systems and machine learning models to mimic expert decision making based on machine learning model predictions

T. **Front End** development refers to static markup, stylization, and light scripting files which could function independently of a chosen server or hosting platform

U. **Back End** development refers to the scripts which allow users and others to interact with the server or with the application in a non-static fashion

V. **Web Application** refers to an application that has its backend files hosted either wholly or partially in an addressable location on a web server