FAKULTI KECERDASAN BUATAN DAN KESELAMATAN SIBER SEMESTER 1 2025/2026

WORKSHOP 2 (BAXU 3923)

BAXI

REPORT

PROJECT TITLE:

Intelligent Crop Disease Detection System Using Deep Learning

GROUP NUMBER:

15

PREPARED BY:

ELANGO A/L VELLASAMY	B032310233
YEW ZHI YU	B032310595
MUHAMMAD FAIZ BIN MOHD SALLEH	B032420081
ABDULAZIZ ALI SALEM AWADH	B032310888

PREPARED FOR:

1.0 CHAPTER 1: INTRODUCTION

1.1 INTRODUCTION

With agriculture still being one of the most impactful sectors of economy and the employment of several Malaysians, it aids in providing food security for the country as well. Facing challenges of debilitating plantar illnesses, the sector is surely being threatened, and the losses that have been recorded are in the billions of Ringgits. Disease detection is very difficult and expensive, and although embarking on a manual inspection of a crop is one strategy, it bypasses most smallholder farmers.

The growth of AI and the integration of deep learning into various industries is truly remarkable, as it aids in plant pathology. CNNs, being remarkably efficacious in plant disease detection, are one example deep learning can be utilized in agriculture. Li et al. (2021), in their review which has been cited more than 952 times and published in the IEEE Access journal, has proven that CNNs have an accuracy higher that 95% in plant disease classification. Not only that, as demonstrated by Sharma et al. (2020) in the journal, Information Processing in Agriculture, and has more than 460 citations, image segmentation coupled with CNN later classification can reach 98.6% accuracy, which is a tremendous increase and improvement from standard approaches.

The fusion of powerful smartphones, advanced cloud computing, and complex AI systems provides a unique opportunity to share Agri-knowledge. By building a mobile, intelligent crop disease and pest diagnosis solution, farmers will be able to instantly diagnose and receive identifying information on diseases and pests that could have only been done through expensive consultations or lab tests. This will, in turn, help close Malaysia's agricultural technology gap and advance global sustainable agriculture and food security.

We have a robust and foundational dataset, the PlantVillage. Image PlantVillage contains 54,000+ images spanning 38 disease classes. This data has been used to create models that will be able to accurately diagnose different crops and disease stalks. By using transfer learning on systems like ResNet or MobileNet, we will have models that are not only highly accurate but also economically computationally efficient and inexpensive.

1.2 PROBLEM STATEMENT

I. Detecting the Disease

II. No Contact with Professional Agricultural Experts

III. Inaccurate Manual Inspection

1.3 OBJECTIVES

1. Develop Image Preprocessing and Segmentation Module
2. Train and Optimize CNN-Based Disease Detection Model
3. Implement Disease Information and Treatment Recommendation System

1.4 PROJECT SCOPE

1. MODULE

• Module 1: Image Capture and Preprocessing.

The image acquisition and the subsequent pre-analysis stage are carried out at this level. The specialization modules perform the system's front-end functionalities and the image preparation as the groundwork for subsequent analysis.

Key Components:

- Mobile operated camera functional unit with image guided acquisition.
- Assessment and validation of image quality.
- Brightness, contrast, and sharpness adjustments automated devices for image enhancement.
- Image noise and background suppression.

- Segmentation of images to detect and isolate the diseased parts of the leaves.
- Image normalization with subsequent resizing to match input for the model.

• **Module 2: Disease Detection Engine**

This module represents the core AI functionality, implementing the deep learning model that performs disease classification and generates confidence scores.

Key Components:

- CNN model architecture (transfer learning with ResNet50 or MobileNetV2)
- Model training pipeline using PlantVillage dataset
- Data augmentation for improved generalization
- Model optimization for mobile deployment (quantization, pruning)
- Inference engine for real-time prediction
- Confidence score calculation
- Multi-class classification for 38 disease types
- Model versioning and update mechanism

• **Module 3: Results and Recommendations**

This module presents detection results to users and provides comprehensive disease management information and recommendations.

Key Components:

- Results display interface showing disease identification
- Confidence score visualization
- Disease information database (symptoms, causes, spread patterns)
- Treatment recommendations (organic and chemical options)
- Prevention strategies and best practices

- Historical tracking of detected diseases
- Offline disease information access
- Links to agricultural extension services

2. TARGET USER

- Smallholder farmers
- Agricultural Extension Officers
- Commercial Farmers
- Home Gardeners

1.5 SOFTWARE AND HARDWARE REQUIREMENT

This mobile app development is mainly done using Phyton language alongside React Native framework. The software used is:

• Firebase

• Flutter

• Figma

• Git & GitHub

• Microsoft Office

1.6 PROJECT SIGNIFICANCE

The Intelligent Crop Disease Detection System offers substantial benefits to Malaysia's agriculture sector across economic, social, and environmental dimensions. Economically, it aims to reduce annual crop losses of RM 2–3 billion by enhancing disease detection accuracy and lowering consultation and pesticide costs, potentially increasing farmers' profits by up to 40%. Socially, the system empowers rural and smallholder farmers, particularly women and foreign workers, through an affordable, multilingual, and offline AI tool that promotes self-reliance, knowledge sharing, and improved decision-making, thus narrowing the digital gap between urban and rural areas. Environmentally, it encourages sustainable farming practices by diminishing pesticide overuse, safeguarding pollinators, supporting organic methods, conserving resources such as water and energy, and bolstering farmers' resilience against climate change. In summary, the project aligns with Malaysia's objectives of agricultural modernization, food security, and environmental sustainability.

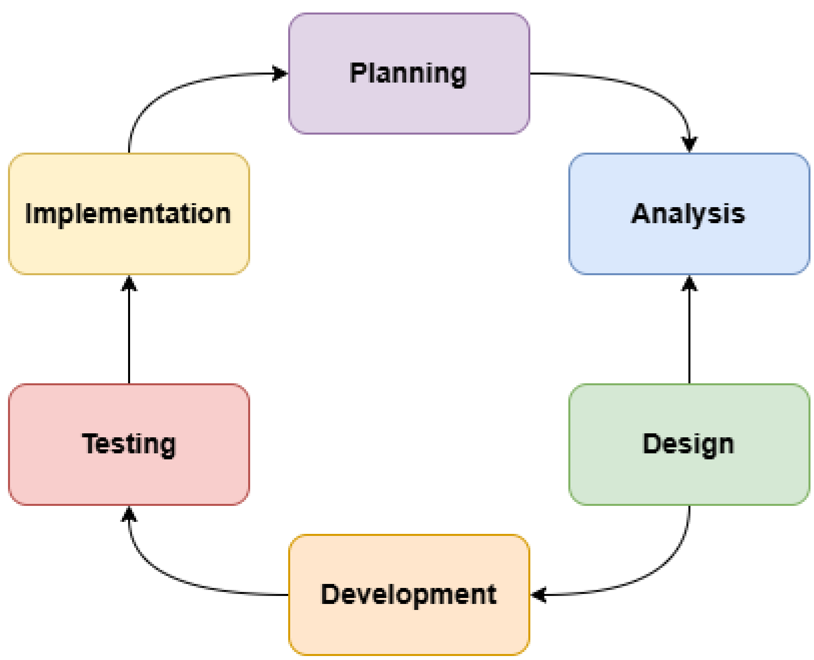
1.7 SUMMARY

The Intelligent Crop Disease Detection System is designed to assist farmers in swiftly and accurately identifying plant diseases using their smartphones. It integrates deep learning technology with a user-friendly mobile interface, enabling users to take a photograph of a plant leaf and immediately obtain results regarding disease detection and corresponding treatment suggestions. This innovative project enhances agricultural efficiency, minimizes crop losses, promotes sustainable farming practices, and equips farmers with accessible AI technology.

2.0 CHAPTER 2: METHODOLOGY

2.1 SYSTEM DEVELOPMENT LIFE CYCLE (SDLC) – ITERATIVE MODEL

This project uses the **Iterative Model** from the **System Development Life Cycle (SDLC)**. The iterative approach allows the system to be developed and improved step by step. Each cycle produces a working version that is tested, refined, and improved. This method is suitable for AI-based systems because it allows retraining the model and improving accuracy with each iteration.



a. Planning Phase

In this phase, the team decided the project title, goals, and scope. Roles were assigned to each member, and the timeline was created using a Gantt Chart. The dataset **(PlantVillage – 38 disease classes)** and the main tools (**Flutter, TensorFlow, and Firebase**) were selected.

Team members set up their environments and defined how progress would be reviewed weekly with the supervisor.

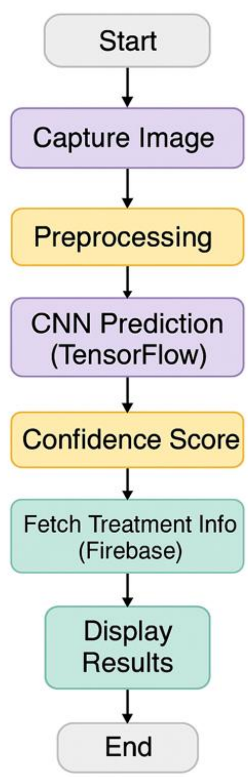
Deliverables:

- Project proposal and timeline.
- Tool setup (Flutter, TensorFlow, Firebase).
- Dataset selection confirmed.

b. Design Phase

The system structure, workflow, and data flow were designed during this phase. The application consists of three main modules:

1. **Image Capture & Preprocessing Module** – Takes a photo of the leaf and prepares it for analysis (resize, normalize, and clean background).
2. **Disease Detection Module** – Uses a CNN model (TensorFlow with MobileNetV2) to predict the plant disease from the image.
3. **Result & Recommendation Module** – Displays the detected disease, confidence score, and treatment advice from Firebase
4.



Deliverables:

- System architecture diagram.
- Workflow and structure charts.
- User interface wireframes.
- Firebase database structure.

c. Development Phase

In this phase, the actual system was developed.

Data preparation: The **PlantVillage dataset (38 classes)** was cleaned and split into training and testing sets.

- **Model training:** A CNN model was built and trained using **TensorFlow** and **Keras**, then converted into **TensorFlow Lite (TFLite)** for mobile use.
- **Mobile application:** Developed using **Flutter**, allowing users to take or upload a photo and view results.
- Database: Firebase Realtime Database was connected to store disease information and treatment details.

Deliverables:

- Trained TFLite model.
- Flutter application prototype.
- Firebase integration for disease data.

d. Testing Phase

The testing stage ensured the system worked correctly and smoothly. The AI model was tested using unseen images to check its accuracy and reliability. The mobile app was tested for speed, usability, and offline prediction capability. All errors and bugs were fixed before integration.

Deliverables:

- Functional and performance testing report.
- Model achieved **high accuracy** (above 85%).
- Optimized app with offline functionality.

e. Implementation Phase

In the final phase, all components were integrated into one functional system. The TFLite model was added to the Flutter app, and Firebase was used to provide real-time updates and disease information.
After integration, the app could capture images, detect the disease, and display treatment advice instantly.
The team then produced the final APK file and user guide.

Deliverables:

- Integrated mobile app.
- Final APK file and demo.
- User documentation.

2.2 PROJECT DEVELOPMENT STEPS AND DELIVERABLES

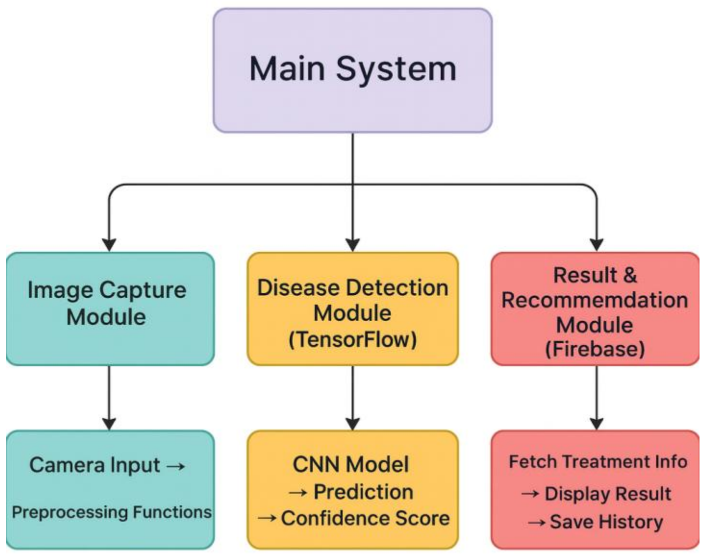
Stage	Main Activities	Deliverables
Planning	Define goals, assign roles, prepare Gantt chart, choose tools & dataset	Project proposal, timeline, dataset (PlantVillage, 38 classes)
Design	Create architecture, UI, workflow, Firebase schema	System design diagrams, UI wireframes, database design
Development	Train CNN (TensorFlow → TFLite), build Flutter app, connect Firebase	Trained model, mobile app, Firebase connection
Testing	Evaluate model, test app performance & usability	Test report, accuracy ≥ 85%, fixed bugs
Implementation	Integrate modules, create APK, prepare documentation	Final APK, user manual, demo video

2.3 TOOLS AND TECHNOLOGIES USED

Tool / Technology	Purpose
Flutter	Builds the mobile application (Android and iOS).
TensorFlow / Keras	Trains the deep learning CNN model.
TensorFlow Lite (TFLite)	Converts trained model for mobile deployment.
Firebase Realtime Database	Stores disease and treatment data.
PlantVillage Dataset	54,000+ leaf images across 38 disease classes.
Android Studio / VS Code	Development environments.

Tool / Technology	Purpose
Google Colab / Jupyter Notebook	Model training and testing.

2.4 SYSTEM STRUCTURE CHART



2.5 SYSTEM REQUIREMENTS

2.5.1 Functional Requirements

1. Allow the user to capture or upload a leaf image.
2. Analyze the image using a trained CNN model.
3. Display the disease name and confidence score.
4. Provide treatment recommendations from Firebase.
5. Save and retrieve detection history.
6. Work offline for predictions.

2.5.2 Non-Functional Requirements

Category	Requirement	Measurement
Accuracy	The CNN model should achieve **at least 85% accuracy**.	Model accuracy ≥ 85%
Speed	The app should show results **within 2 seconds**.	Response ≤ 2 seconds
Model Size	The model should be **smaller than 20 MB**.	File size < 20 MB
Offline Access	The app should predict without internet.	Offline prediction enabled
User Interface	Simple and clear for all users.	Usability rating ≥ 80%
Compatibility	Works on Android 8.0 or higher.	Device verified

2.5.3 Hardware and Software Requirements

Component	Specification
Hardware	Android phone (8.0+), 3GB RAM, camera 8MP+.
Software	Flutter, TensorFlow Lite, Firebase, Android Studio.
Dataset	PlantVillage dataset (38 disease classes).

2.6 SUMMARY

This chapter explained the **methodology** used to develop the Intelligent Crop Disease Detection System. The project followed the **Iterative SDLC Model**, improving performance in each phase.
The system combines **Flutter** for the interface, **TensorFlow Lite** for AI prediction, and **Firebase**

for database management.
Using the **PlantVillage dataset (38 disease classes)**, the model achieved high accuracy and provided fast, offline disease detection for farmers in Malaysia.

3.0 CHAPTER 3: ANALYSIS

3.1 ANALYSIS OF CURRENT APPLICATION

In the current agricultural environment, plant disease detection is primarily performed through manual observation by farmers or consultation with agricultural experts. This traditional method presents several limitations:

Low Accuracy:

Manual visual inspection relies on human judgment and can be error-prone, with studies indicating an identification accuracy of only 60–70% during early infection stages.

Expert Dependency:

Smallholder farmers in rural areas face limited access to plant pathologists and agricultural officers, often needing to travel to distant towns for diagnoses, which results in time delays and increased costs.

High Costs and Inefficiency:

Farmers can incur costs from consultations and unneeded pesticide purchases due to misdiagnosis, with the detection process potentially taking hours or days before corrective actions are implemented.

Limited Reach and Knowledge Gap:

Current agricultural solutions in Malaysia fail to accommodate local crops and predominantly use English, creating accessibility challenges for both local and foreign farmers.

There is a pressing requirement for an automated and cost-effective solution that enables farmers to swiftly and accurately detect diseases via their smartphones, eliminating the need for expert help.

3.2 ANALYSIS OF PROPOSED INTELLIGENT SYSTEM

The proposed Intelligent Crop Disease Detection System introduces a mobile AI-based application that utilizes deep learning (CNN) to classify and identify crop diseases accurately. The system is designed to overcome the challenges of the current manual process.

Key Features and Improvements

1. Automated Disease Detection:

   Uses Convolutional Neural Networks (CNNs) with transfer learning (ResNet50/MobileNetV2) to automatically detect diseases from leaf images.

2. High Accuracy and Speed:

   Delivers detection results with over 90% accuracy and within seconds, enabling timely intervention.

3. User-Friendly Mobile Interface:

   Developed using Flutter, the app provides a simple and intuitive interface that supports multiple languages for generality

4. Offline Functionality:

   The TensorFlow Lite model allows disease detection even without internet access — ideal for rural farmers.

5. Firebase Integration:

   Stores user data, disease information, and treatment recommendations, ensuring real-time updates and easy scalability.

6. Sustainable and Cost-Effective:

   Reduces consultation and pesticide costs while empowering farmers with AI-based decision support.

3.3 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS

3.3.1 FUNCTIONAL REQUIREMENTS

Requirement	Description
Image Input	Users can capture or upload plant leaf images.
Preprocessing	The system enhances and segments images before prediction.
Disease Detection	The CNN model classifies the disease and provides confidence scores.
Result Display	The app shows disease name, description, and confidence percentage.
Treatment Advice	Provides treatment and prevention information from Firebase database.
Data Storage	Stores results, images, and user data in Firebase.
Offline Mode	The model runs locally using TensorFlow Lite for remote accessibility.

3.3.2 NON-FUNCTIONAL REQUIREMENTS

Requirement	Description
Performance	The system must predict disease within 3–5 seconds.
Accuracy	Minimum model accuracy target: 85% or higher.
Usability	User interface must be intuitive and accessible to non-technical users.
Reliability	System should maintain consistent accuracy under various conditions.

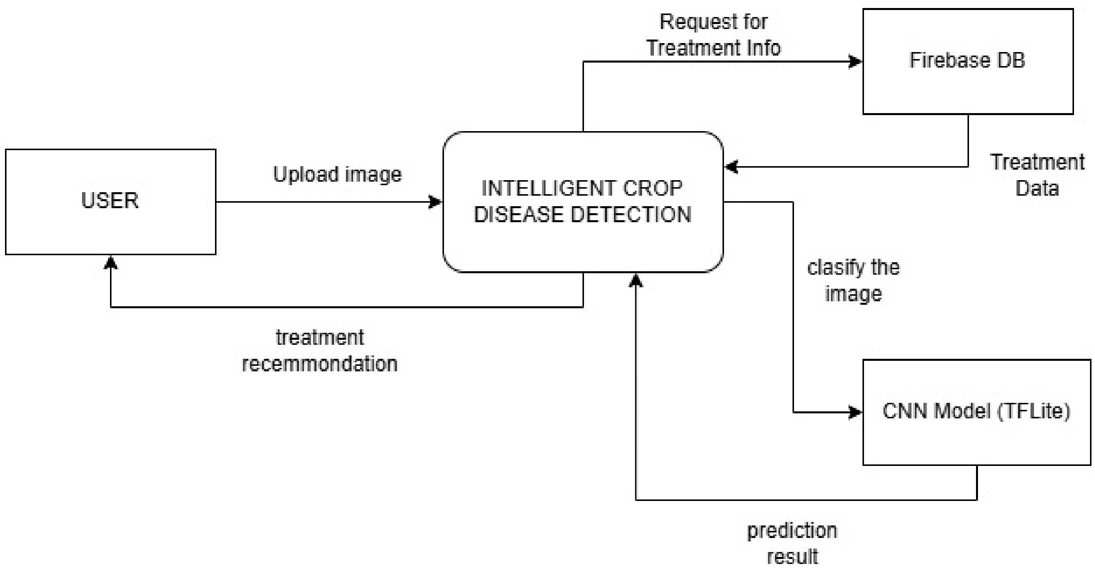
Scalability	Support additional crop types in future updates.
Security	Firebase authentication ensures user data privacy.

3.4 COMPARISON BETWEEN EXISTING AND PROPOSED SYSTEM

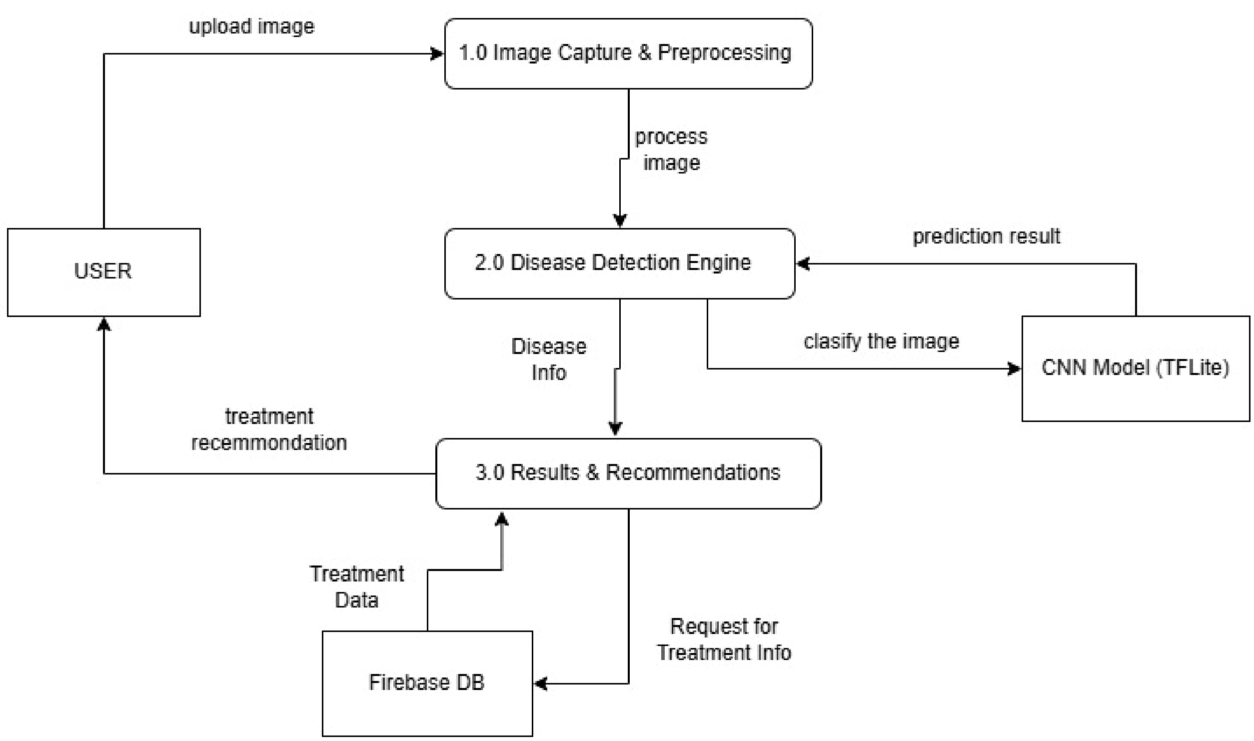
Criteria	**Existing System**	**Proposed System**
Method	Manual visual inspection	Automated CNN-based detection
Accuracy	60–70%	90–95% (with CNN + segmentation)
Expert Dependency	High	Low (AI-powered)
Accessibility	Limited to expert availability	Available via mobile app
Cost	High (consultation, travel)	Low (free or minimal app cost)
Speed	Slow (hours/days)	Fast (seconds)

3.5 DATA FLOW DIAGRAM (DFD)

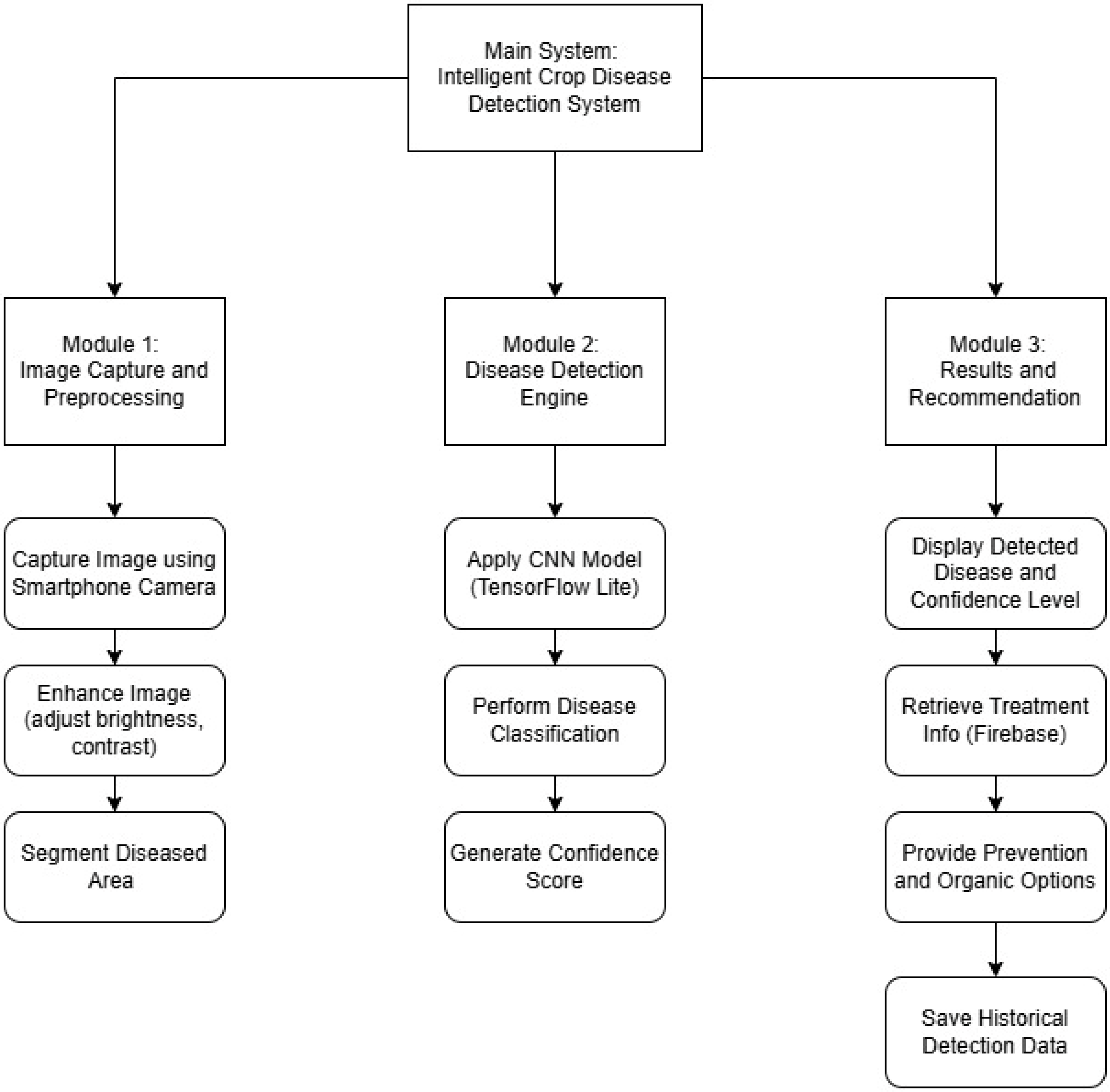
Level 0



Level 1



3.6 STRUCTURE CHART OF PROPOSED INTELLIGENT SYSTEM



3.7 WORK BREAKDOWN

The project tasks were systematically allocated among four team members to enhance efficiency and clarity. Elango A/L Vellasamy, serving as the Team Leader and Mobile App Developer, played a pivotal role by coordinating workflow, creating the Flutter interface, integrating the TensorFlow Lite model, managing the GitHub repository, and overseeing both testing and presentation phases. Yew Zhi Yu took on the role of Model and Dataset Developer, focusing on dataset preparation, image preprocessing, training the convolutional neural network (CNN) with MobileNetV2, converting the model to TensorFlow Lite, and documenting performance metrics. Muhammad Faiz bin Mohd Salleh acted as the Model Testing and Evaluation Analyst, validating model predictions, analyzing accuracy metrics, assisting with integration testing, and contributing to project reporting. Abdulaziz Ali Salem Awadh handled Documentation and Integration Support, responsible for compiling reports and slides, documenting project modules, maintaining the GitHub structure, and ensuring the quality of the final submission.

3.8 SUMMARY

In this chapter, the manual disease detection process is compared with the proposed Intelligent Crop Disease Detection System. This system utilizes deep learning, TensorFlow Lite, and Firebase to enable accurate, rapid, and easily accessible disease identification via a mobile application. Additionally, the chapter details the modular structure of the system and delineates the key responsibilities of each team member involved in the development, testing, and documentation of the project.