



Papaya Buddy

Enhancing Papaya Cultivation Through Advanced Monitoring

24-25J-069

Introduction

- Papaya is a vital crop in tropical agriculture, contributing significantly to both the economy and nutrition. It covers extensive cultivation areas and provides essential income for many farmers.
- The high incidence of pests and diseases poses significant threats to papaya production and farm productivity. Early detection and management of these diseases are critical to sustaining healthy crops and maximizing yields.
- A smart agricultural solution with automated plant disease detection and classification tools offers a valuable resource for supporting farm decision-making. Our system integrates advanced Deep Learning technologies to diagnose papaya plant diseases accurately and efficiently, even those undetectable by the untrained human eye.

Research Question

- Many diseases are difficult to detect in their early stages, leading to severe outbreaks and significant crop losses.
- Most farmers are not aware of the specific diseases and pests affecting papaya cultivation. They only recognize that their plants are diseased without understanding the exact nature of the problem.
- Due to this lack of knowledge, farmers often resort to using any available chemical treatments or removing affected plants entirely, leading to reduced income and yield.



How to Identify ?

- Papaya Mosaic and Cercospora Leaf diseases
- Papaya Ringspot Virus and Powdery Mildew Fruit diseases
- Mite and Mealy Bugs
- Maturity level of papaya fruit

Main & Sub Objectives

Main Objective

Advanced Monitoring system for disease and pest controlling for papaya cultivation.

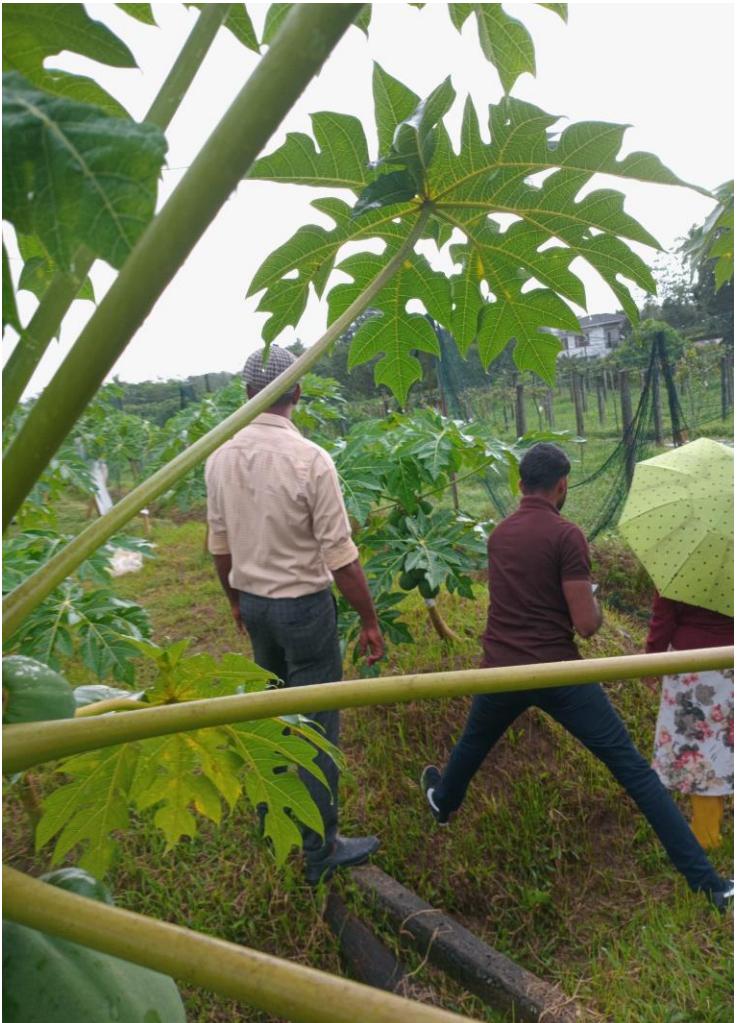
Sub Objectives

- 1** Identification and classification of Papaya Mosaic and Cercospora leaf disease.
- 2** Identification and classification of Papaya Ringspot Virus and Powdery Mildew fruit disease.
- 3** Identification and classification of Mite and Mealy Bug.
- 4** Identification and Classification of maturity level of papaya fruit

Field Visits



Field Visit – FRID Horana



Diwulapitiya Farm

Infected Plants
in early stage



After 4 months



The plants are
infected with
diseases

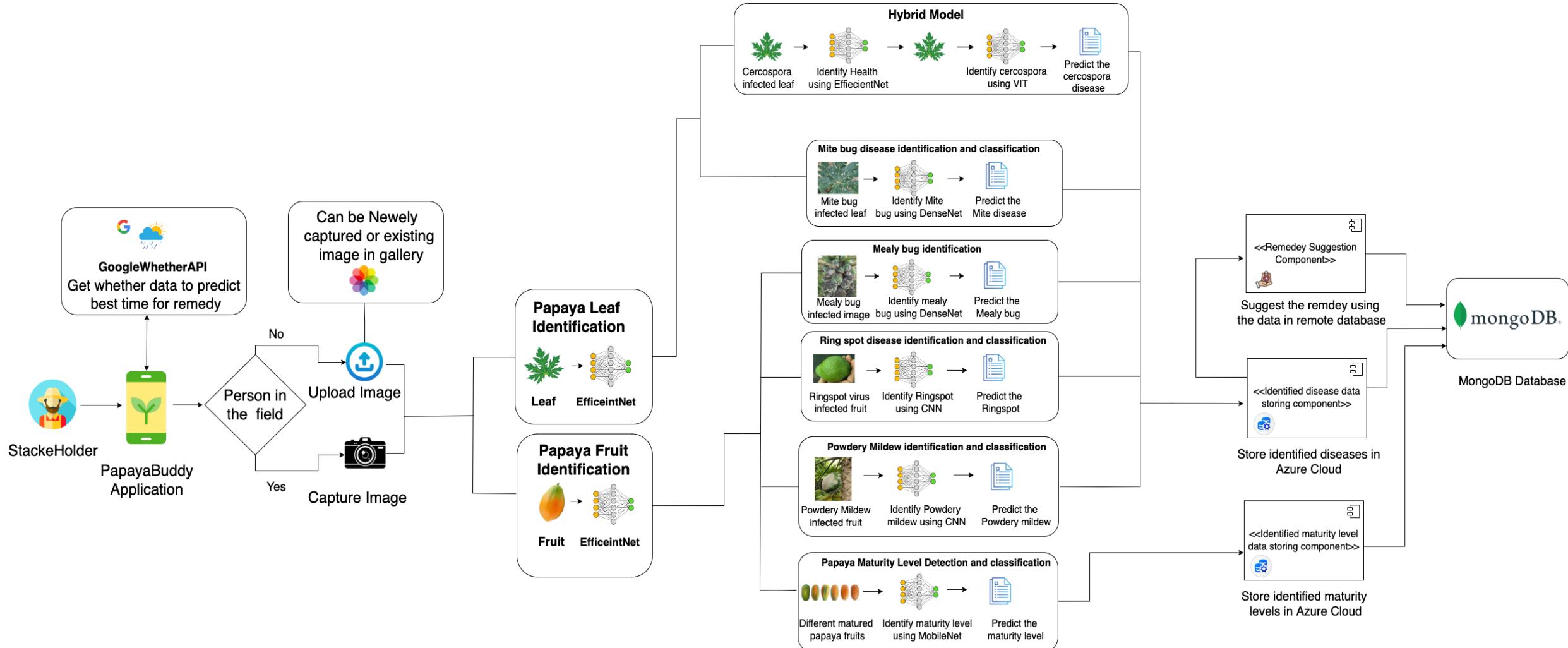


4th Field Visit - Yield Loss - Diwulapitiya



System Diagram

Enhancing Papaya Cultivation through Advanced Monitoring (Papayabuddy)



Why we selected Microservices Architecture

- **Dependency Conflicts**

Different models require different libraries, leading to version mismatches.

- **Scalability Issues**

Scaling one model requires scaling the entire system, increasing costs.

- **Single Point of Failure**

If one component fails, the entire application may become unresponsive.

- **Technology Flexibility**

Each microservice can be developed using the most suitable programming language And framework, allowing teams to work independently without constraints.

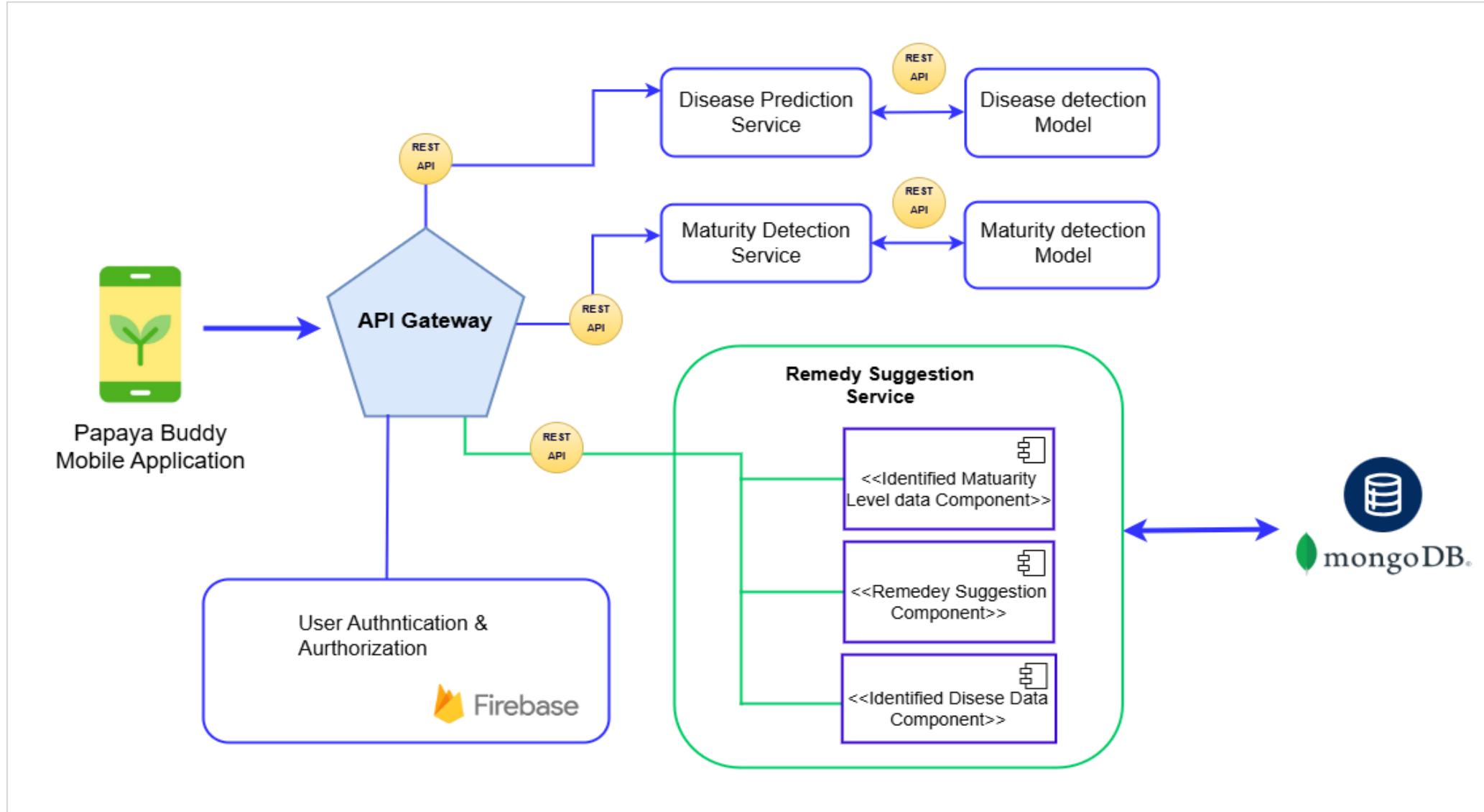
Dockerization Issue

The screenshot shows the Docker Desktop application interface. The left sidebar has navigation links: Containers, Images (which is selected and highlighted in blue), Volumes, Builds, Docker Hub, Docker Scout, and Extensions. The main area is titled "Images" with a "Local" tab selected. It displays 8 images with the following details:

Name	Tag	Image ID	Created	Size	Actions
pms-image	latest	adcdcf2c14061	4 days ago	655.89 MB	View Edit Delete
backend-api-gateway	latest	a0604a6afa7c	4 days ago	540.05 MB	View Edit Delete
backend-users-service	latest	90b7f75c74d3	4 days ago	219.88 MB	View Edit Delete
papaya-disease-classifier	latest	22d55e4984e9	9 days ago	8.8 GB	View Edit Delete
gcr.io/k8s-minikube/kicbase	<none>	fd2d445ddcc3	2 months ago	1.85 GB	View Edit Delete
kong/kong-gateway	latest	cb838b4090cf	3 months ago	655.87 MB	View Edit Delete
postgres	16.4	e62fbf9d3e2b	6 months ago	610.91 MB	View Edit Delete
dpage/pgadmin4	8.11	822dd7a1fa68	7 months ago	722.16 MB	View Edit Delete

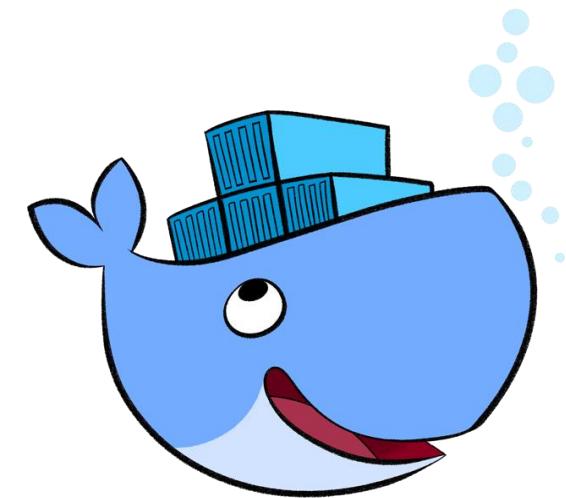
A yellow box highlights the row for "papaya-disease-classifier". The bottom status bar shows "Engine running", system resources (RAM 3.02 GB, CPU 1.02%, Disk 30.93 GB used / limit 1006.85 GB), and notifications for a new terminal and a new version available.

Microservices Architecture



System Architecture

- 📱 Flutter Mobile App – User-friendly interface for disease detection, ripeness classification, and remedy suggestions.
- 🔒 Firebase (User Management & Authentication) – Handles user authentication, role-based access control, and real-time sync.
- 📁 Node.js (Backend Service) – Fetches and stores remedy data, disease information, and maturity data.
- 🤖 Fast API (Model Hosting) – Load CNN models and disease prediction
- 🌐 Kong API Gateway – Provides authentication, authorization, and load balancing, ensuring security and efficient request handling.
- 🚀 **Microservices with Firebase ensure a secure, scalable, and high-performance system!**



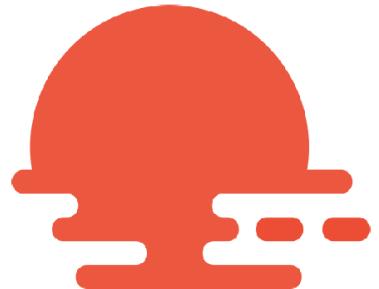
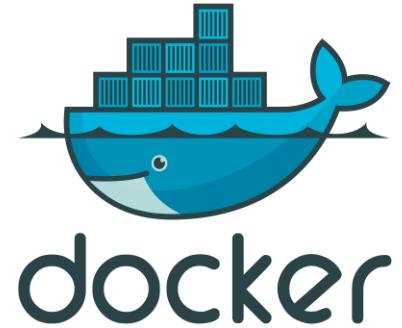
Used Technologies



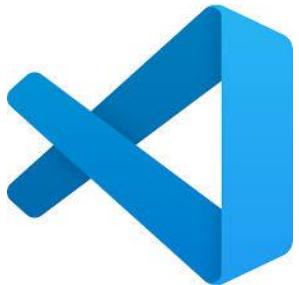
Flask



POSTMAN



FastAPI



Coding Best Practices

- **Write Clean & Readable Code**
Maintain clarity and structure for better collaboration.
- **Modular Design**
Break code into smaller, reusable functions for efficiency.
- **Meaningful Documentation**
Add concise comments to enhance code understanding.
- **Version Control & Agile Workflow**
Use GitHub with a proper branching strategy and a Kanban board for streamlined project management.
- **Continuous Improvement**
Stay updated and refine skills through learning.
- **Follow DRY Principle**
Avoid redundancy by reusing code effectively.



Git Hub Repository

Code and resources for the project, available for collaboration and implementation

The screenshot shows a GitHub repository page for 'Papaya-Buddy'. The repository is public and has 36 branches and 0 tags. It contains 255 commits from user 'IT21161056'. The commit history includes changes to '.vscode', 'backend', and 'mobile_app' folders, as well as files like 'MobileNet.ipynb', 'README.md', and 'Sequential.ipynb'. The 'About' section describes it as a research project with 0 stars, 1 watcher, and 0 forks. The 'Releases' section indicates no releases have been published. The 'Contributors' section lists 5 contributors. The README file is present but empty.

Papaya-Buddy (Public)

development 36 Branches 0 Tags

IT21161056 Merge pull request #124 from IT21161056/diagnosis-list-integration 1dc428c · 8 hours ago 255 Commits

.vscode added 2 dependency http & image picker last month

backend prediction history view 8 hours ago

mobile_app fix:build issue 8 hours ago

MobileNet.ipynb added trained models 3 months ago

README.md folder structure added for the frontend last month

Sequential.ipynb added trained models 3 months ago

README

Papaya-Buddy

Research Project (IT4010)

After clone: flutter pub get

About

research project

- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

Report repository

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Contributors 5

SLIIT FACULTY OF COMPUTING

Kanban Dashboard

For project management to flow smoothly and meet the deadline

The screenshot shows a Kanban dashboard for the 'Papaya Buddy' project on GitHub. The dashboard is organized into five columns representing different stages of the project:

- Backlog**: 1/5 items, Estimate: 0. Status: This item hasn't been started. Item: Papaya-Buddy #10 Create FastApi.
- Ready**: 1 item, Estimate: 0. Status: This is ready to be picked up. Item: Papaya-Buddy #42 Style Diagnosis UI.
- In progress**: 4/3 items, Estimate: 0. Status: This is actively being worked on. Items: Papaya-Buddy #7 Dashboard UI, Papaya-Buddy #9 Image Capturing & Upload, Papaya-Buddy #8 Create image selection UI, Papaya-Buddy #35 Prediction details display UI.
- In review**: 1/5 items, Estimate: 0. Status: This item is in review. Item: Papaya-Buddy #44 healthy view ui creation.
- Done**: 12 items, Estimate: 0. Status: This has been completed. Items: Papaya-Buddy #25 Create Treatments view UI, Papaya-Buddy #22 Change Sign-up UI styles and add input validation, Papaya-Buddy #6 Sign-up UI, Papaya-Buddy #5 Sign-In UI, Papaya-Buddy #14 Create a Sample Splash screen, Papaya-Buddy #16 Disease view UI, Papaya-Buddy #13.

Each card in the backlog and ready columns includes a 'P2' priority indicator. The 'In progress' column has a count of 4/3, likely indicating incomplete items. The 'Done' column has a higher estimate of 12 compared to the other columns.

Postman Collection

Postman collection for API testing and documentation.

The screenshot shows a Postman collection interface for the PapayaBuddy mobile application. The top navigation bar includes the Postman logo, environment dropdown (No Environment), layout dropdown (Double Column), language dropdown (cURL - cURL), and a public sharing link. A note at the top states: "This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page."

PAPAYABUDDY **PapayaBuddy**

Introduction

- > Disease
- > Maturity
- > Treatment
- > Upload

Disease

PATCH update disease

http://localhost:5080/api/v1/disease/67d3f00ab5823abff3f8622d

Body raw (json)

```
{  
  "name": "Mosaic"  
}
```

Example Request

New Request ▾

```
curl  
curl --location --request PATCH 'http://localhost:5080/api/v1/disease/67d3f00ab5823abff3f8622d' \  
--data '{  
  "name": "Mosaic"  
}'
```

Example Response

Body Headers (8) 200 OK

```
{  
  "success": true,  
  "message": "Disease \"Mosaic\" updated successfully",  
  "data": {  
    "_id": "67d3f00ab5823abff3f8622d",  
    "name": "Mosaic",  
    "affected_area": "leaves",  
    "symptoms": [  
      "Yellowing of leaves",  
      "Mottling of light and dark green on leave",  
      "Leaf curling and distortion"  
    ]  
  }  
}
```

GET get-all

Example Request

New Request ▾

MongoDB Schema

MongoDB database setup and schema for the project.

The screenshot shows the MongoDB Compass interface. The left sidebar navigation bar includes 'Atlas', 'SE Project gr...', 'Access Manager', 'Billing', 'Clusters' (selected), 'Data Services' (highlighted in green), 'Charts', 'DATABASE' (selected), 'diseases' (selected), and 'histories', 'maturities', 'suggested_images', 'treatments'. The main panel displays the 'papaya_buddy_db.diseases' collection with 8 documents. The first document is shown in detail:

```
_id: ObjectId('67d54600dfaef44e6aab10963')
name : "Mite"
affected_area : "leaves"
symptoms : Array (4)
disease_type : "Bug"
description : "Mite bugs are tiny arachnids that feed on plant sap, causing damage su..."
preventive_measures : "Prevent mite infestations by regularly spraying plants with neem oil o..."
suggested_image_urls : Array (3)
created_at : 2025-03-15T09:18:56.136+00:00
__v : 0
```

The second document listed is:

```
_id: ObjectId('67d54750dfaef44e6aab10965')
name : "Mealy"
```

Firebase

Firebase user management for authentication and access control

The screenshot shows the Firebase console interface for managing authentication users. The left sidebar includes links for Project Overview, Authentication (which is selected), Firestore Database, Storage, Genkit, Vertex AI, Product categories, Build, Run, Analytics, AI, and All products. The main content area is titled 'Authentication' and shows a table of users. A banner at the top right encourages enabling Multi-factor Authentication (MFA) by May 13, 2025.

Authentication

Users Sign-in method Templates Usage Settings Extensions

The following Authentication features will stop working when Firebase Dynamic Links shuts down on August 25, 2025: email link authentication for mobile apps, as well as Cordova OAuth support for web apps.

Identifier	Providers	Created	Signed In	User UID
amal@gmail.com	✉️	Mar 18, 2025	Mar 18, 2025	lh3S0dPK6LVPj2nfVliXkTh7rd...
kamal@gmail.com	✉️	Mar 13, 2025	Mar 13, 2025	PGiTuCdNuDQ4QoHVvGBrNtL...
anoj@gmail.com	✉️	Mar 13, 2025	Mar 18, 2025	tC1HdBRcvhYtPryNppjFTFtHF...
vikum@gmail.com	✉️	Mar 12, 2025	Mar 18, 2025	CbpeWl3KPue25bAn6VEH1Xr...

Rows per page: 50 1 - 4 of 4

Figma UI Creation

The screenshot displays the Figma interface for a mobile application named "PapayaBuddy". The left sidebar shows the project structure with pages like "Anoj", "Pasindu", and "Vikum". The main canvas displays ten mobile screens (iPhone 16 models) arranged in two rows. The top row includes screens for "Cure Your Crop", "Treatments", "Fruit Disease Detection", "Ring Spot Virus", "Ask Community", and "Maturity Stage". The bottom row includes screens for "Healthy Fruit", "Mite Bug", "Cercospora", "Maturity Detection", and "Papaya Maturation Guide". Each screen shows various components such as cards for diseases, symptoms, treatment instructions, and user interaction elements like "Sign In", "Save to Diagnoses", and "Predict". The right sidebar shows design, prototype, and export tools.



IT21161056 | Peiris M. M. A. E

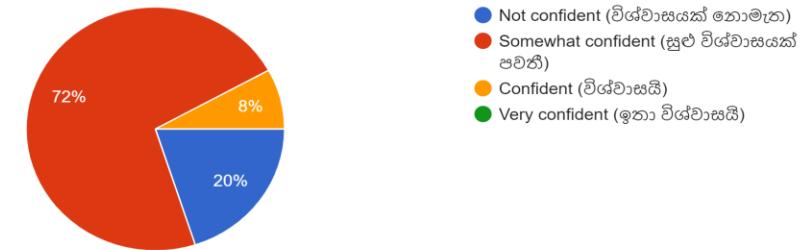
Software Engineering

"How can we revolutionize papaya farming in Sri Lanka?"

Leveraging cutting-edge mobile technology and machine learning,
to precisely detect and manage:

- Papaya Mosaic Virus
- Cercospora Leaf Spot

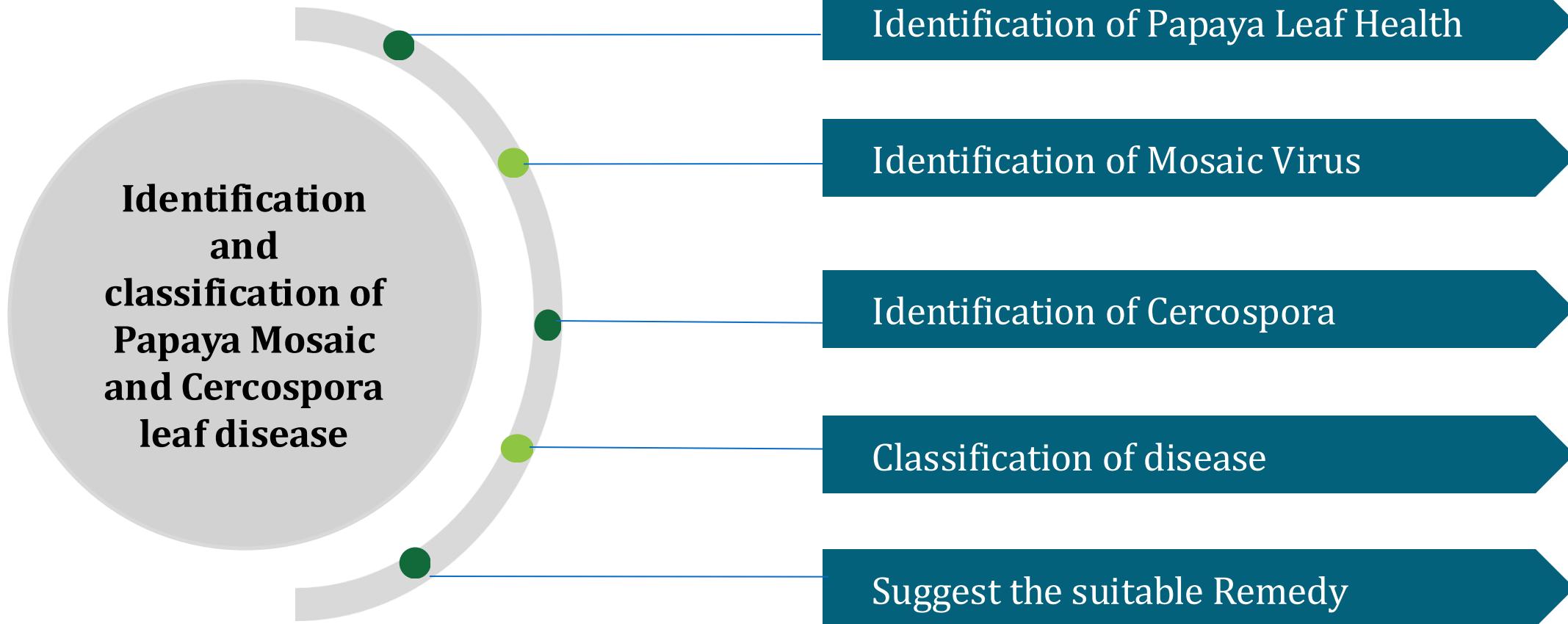
9. How confident are you in your ability to identify diseases in your papaya plants? (ඔබේ පැශේෂල් පැලට රෝග හඳුනා ගනීමට ඔබට ඇති නැඹුමාව ගැන ඔබට කෙනරම් විය්වාසදු?)
25 responses



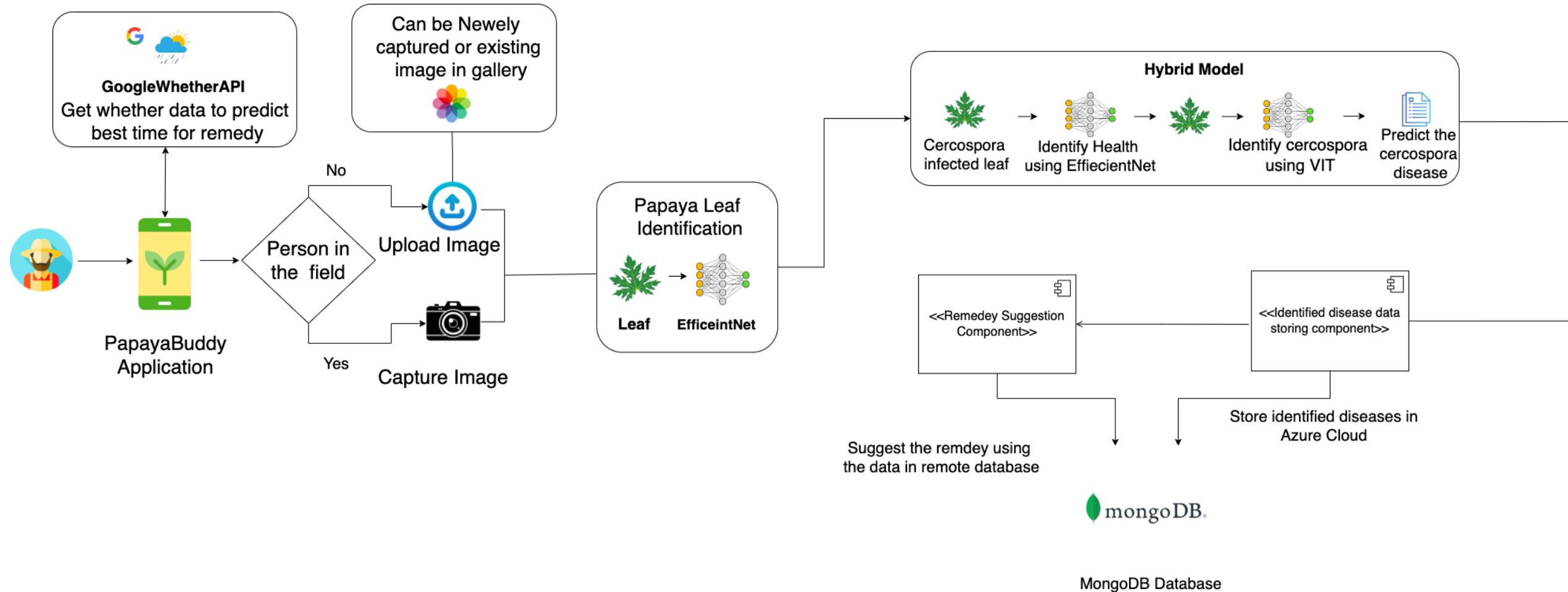
Can this innovation empower rural farmers with real-time, accurate disease diagnosis and effective management strategies?

Specific and Sub Objectives

Introduction



System Diagram



Evidence of Completion



Data Collection
& Pre-Processing



Data Augmentation



Data Visualization



Identify Healthy & Unhealthy
Leaves - EfficientNet



Identify Cercospora &
Mosaic Virus - ViTs



Create Hybrid Model



Fast API for call Hybrid model



Display the results on the
Mobile application

Data collection

Methodology

Cercospora



Mosaic



Healthy



Class Name	Data set
Healthy	339 Images
Cercospora	429 Images
Mosaic	250 Images

Data Augmentation & Preprocessing

- Preprocessed to a size of 224 x 224, making it best suited for model training.

```
IMG_SIZE = 224
transform = transforms.Compose([
    transforms.RandomResizedCrop(IMG_SIZE, scale=(0.8, 1.0), ratio=(0.75, 1.33),
        interpolation=transforms.InterpolationMode.BICUBIC), # Crop the best area
    transforms.RandomHorizontalFlip(p=0.5), # Flip with 50% probability
    transforms.RandomRotation(degrees=5, expand=False), # Restrict rotation
    transforms.ColorJitter(brightness=0.1, contrast=0.1, saturation=0.1, hue=0.02), # Minor color adjustments
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) # Standard normalization
])
```

Hybrid Inference

Model Name	google/vit-base-patch16-224
Pretrained	Yes
Data set Size	669 Images
Classes	2
Batch Size	Batch size = 32
Image Size	224×224
No of Parameters	86.6M params
Model Size	343 MB ~
Used Memory/ RAM	Tesla T4 GPU
Used Environment	Google Collab



Combine Models

Model Name	efficientnet_b0
Pretrained	Yes
Data set Size	761 Images
Classes	2
Batch Size	Batch size = 32
Image Size	224×224
No of Parameters	5.33M params
Model Size	16 MB ~
Used Memory/ RAM	Tesla T4 GPU
Used Environment	Google Collab

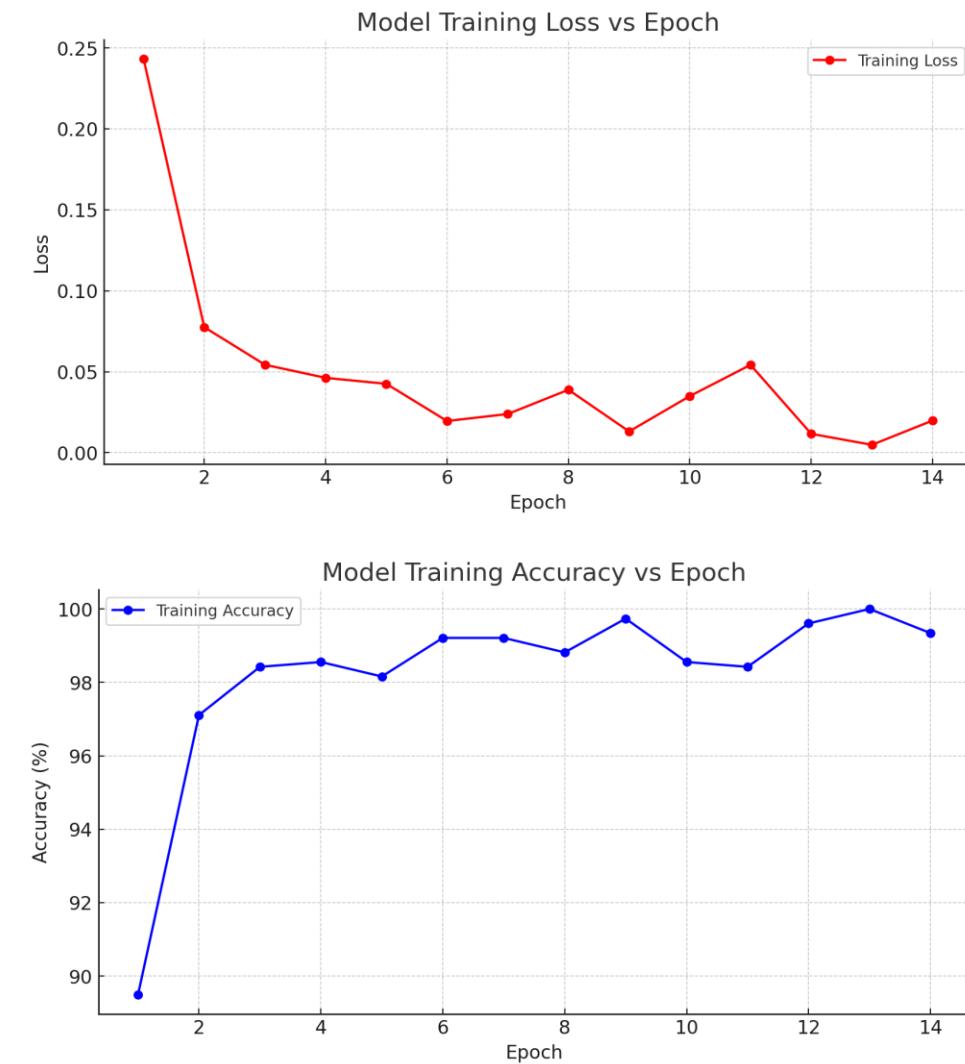
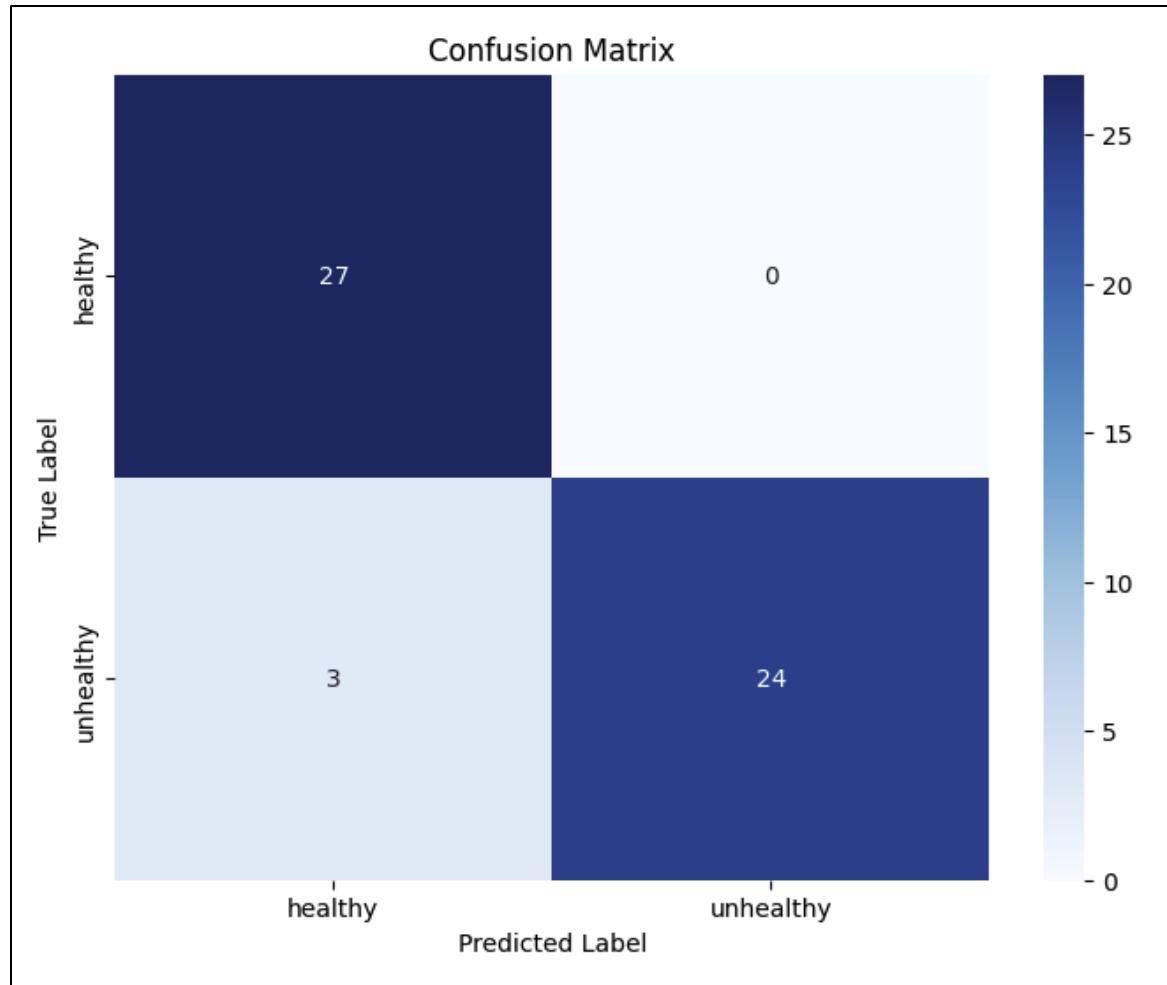
Hybrid Inference

```
class HybridModel(nn.Module):
    def __init__(self, efficientnet_model, vit_model):
        super(HybridModel, self).__init__()
        self.efficientnet = efficientnet_model
        self.vit = vit_model

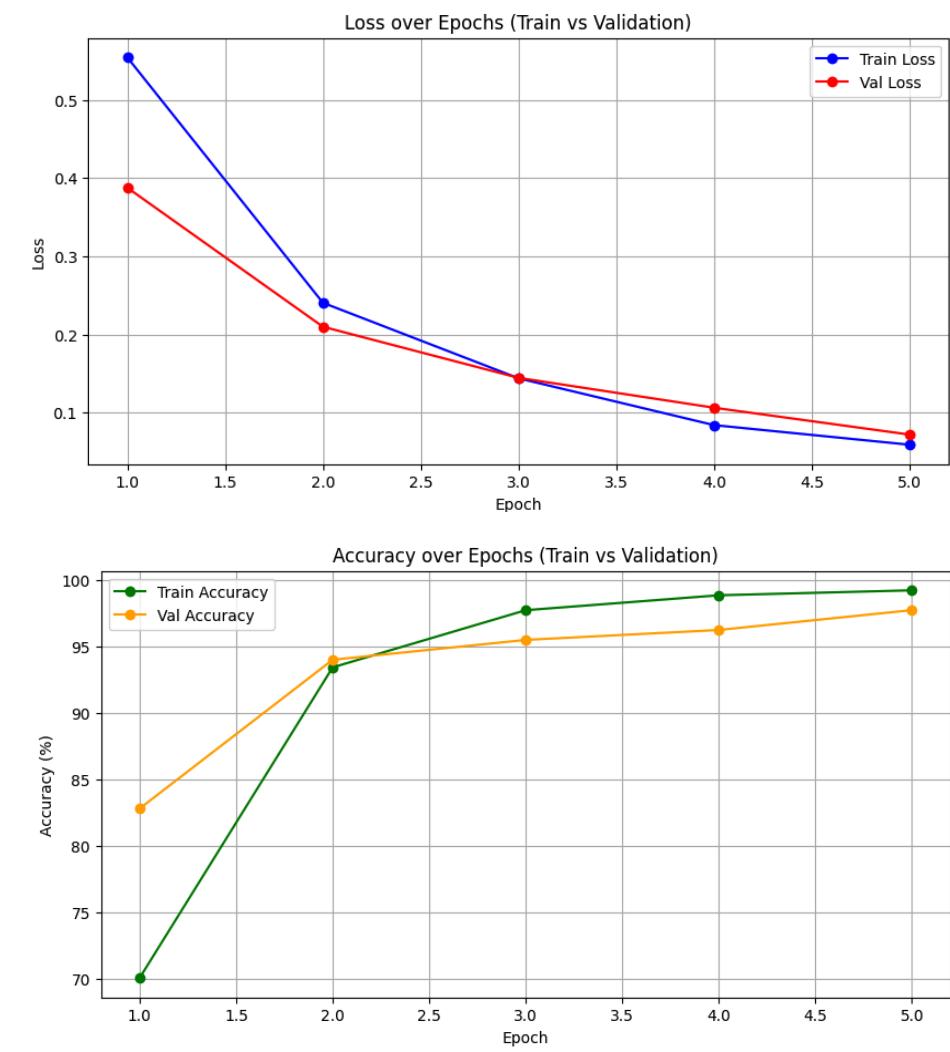
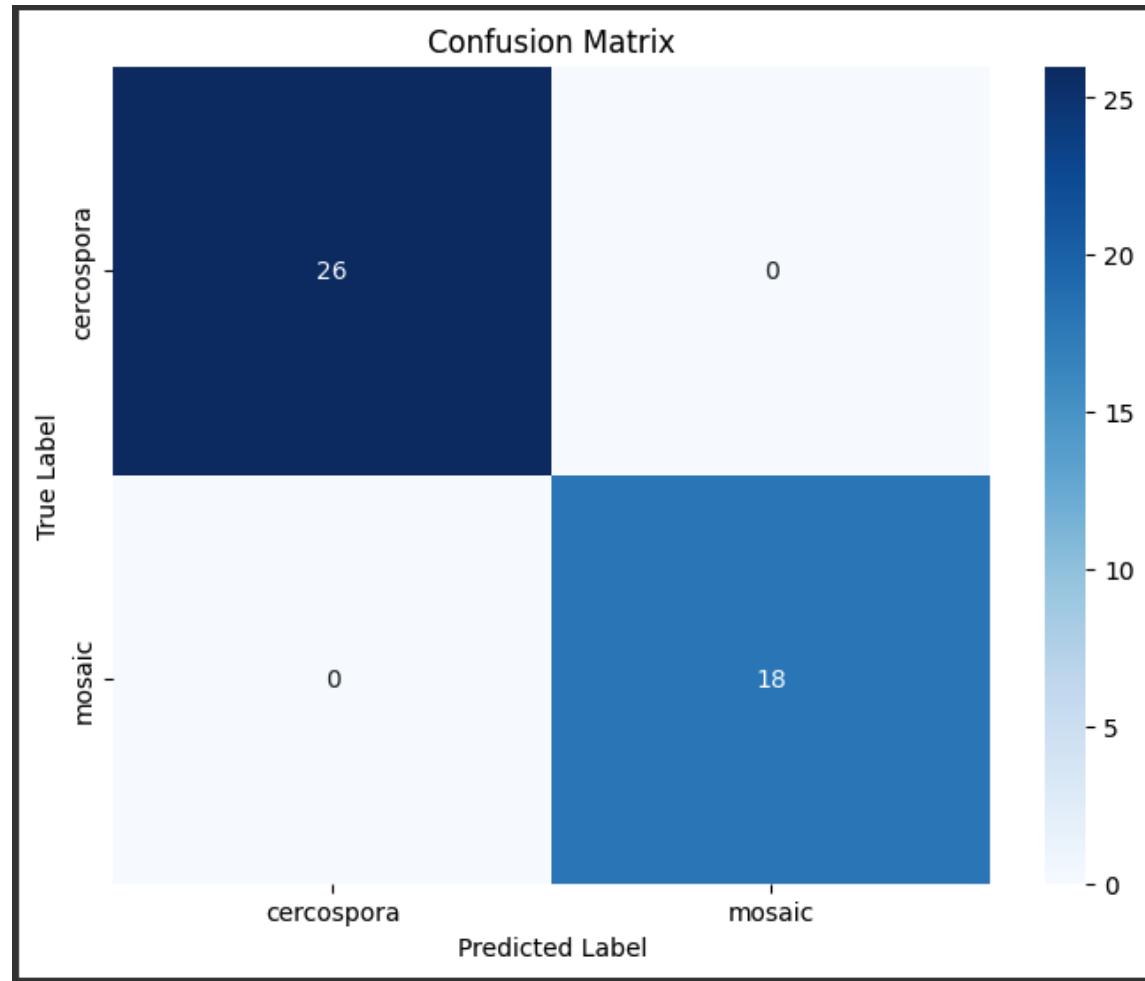
    def forward(self, x):
        eff_output = self.efficientnet(x)
        _, eff_pred = torch.max(eff_output, 1)

        if eff_pred.item() == 1:
            vit_output = self.vit(x).logits
            _, vit_pred = torch.max(vit_output, 1)
            return eff_pred.item(), vit_pred.item()
        else:
            return eff_pred.item(), None # Healthy, so no disease classification
```

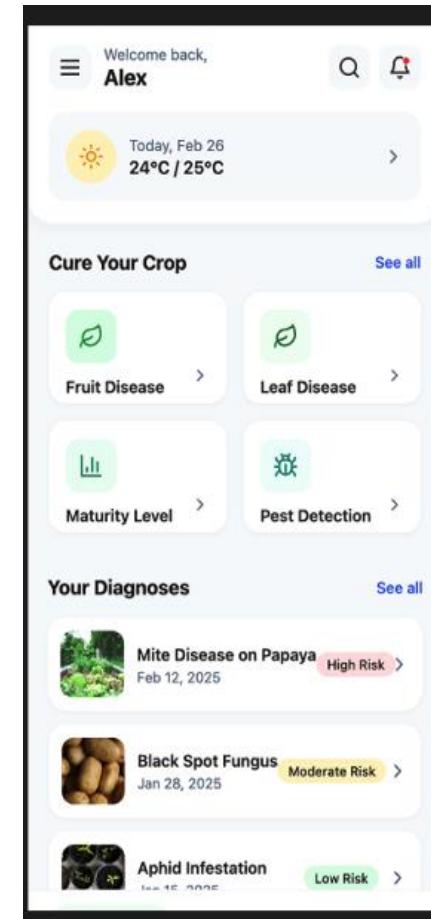
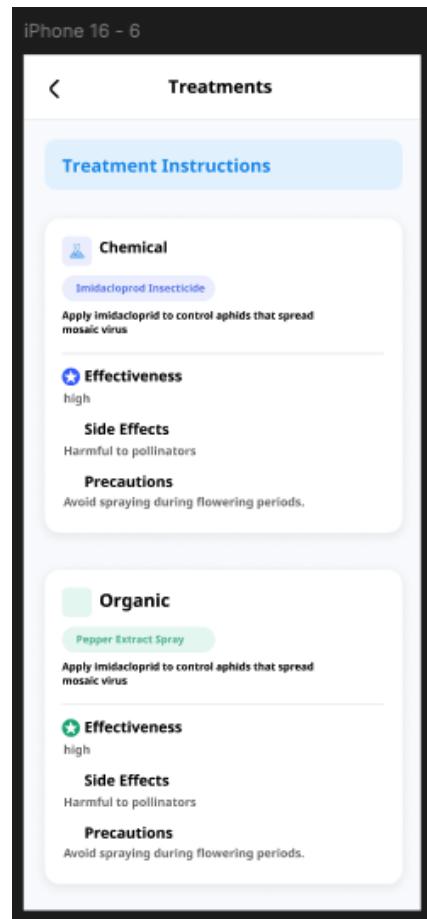
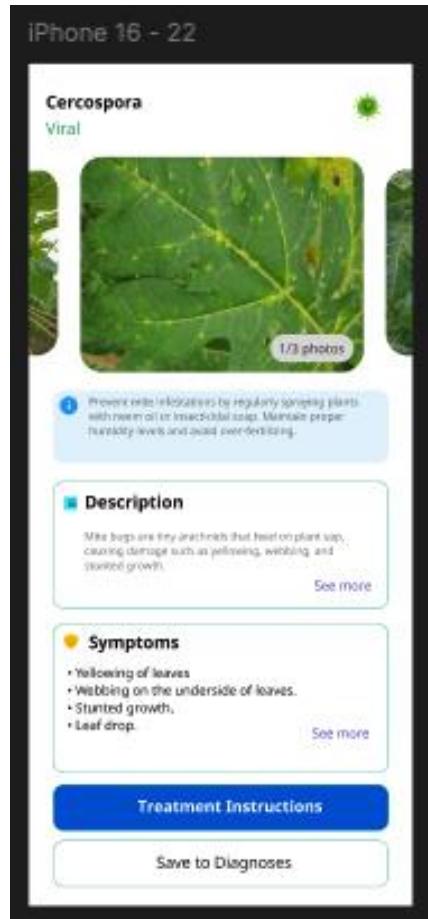
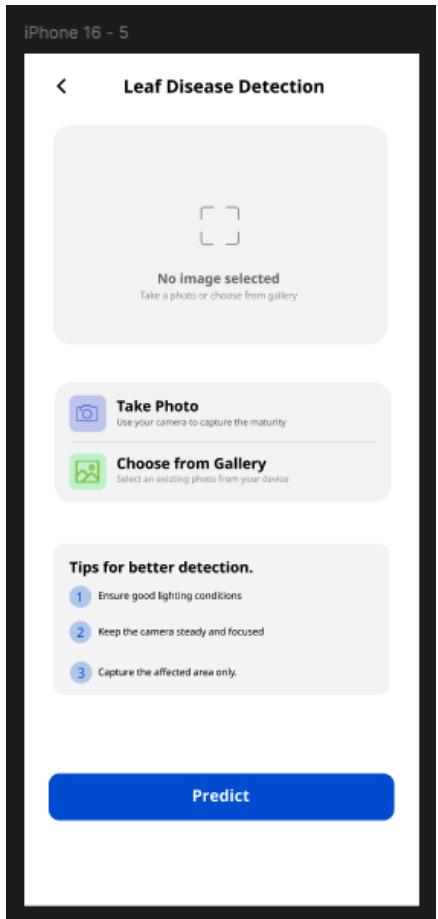
EfficientNet Performance



Vit Model Performance



Mobile Application Mockups



Mobile Application UIs

The image displays five screenshots of a mobile application interface, likely for a plant disease detection app.

- Leaf Disease Detection (Left):** Shows a camera viewfinder with a green leaf. Below it are two buttons: "Take Photo" and "Choose from Gallery".
- Leaf Disease Detection (Middle):** Shows a green leaf with several brown spots. Below it are two buttons: "Take Photo" and "Choose from Gallery".
- Prediction:** Shows the detected disease as "Cercospora fungal". It includes a thumbnail of the leaf image and a note: "Ensure proper air circulation by pruning plants, avoid overhead watering to reduce leaf moisture, apply fungicides if necessary, remove infected plant debris, and practice crop rotation to prevent fungal buildup."
- Treatments:** Shows treatment instructions for "Copper-based Fungicide". It includes sections for "Effectiveness" (high), "Side Effects" (can be toxic to beneficial microbes), and "Precautions" (use protective gear during application).
- Profile:** Shows a user profile for "Anoj Peiris", described as a "Plant enthusiast & organic farmer". It includes contact information: Email (anoj@gmail.com), Phone (0710624368), and City (Malabe). It also has account settings for Push Notifications, Location Services, and Analytics, all turned on.



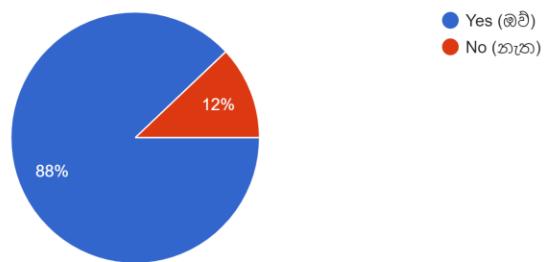
IT21160066 | K.K.P. Perakum

Software Engineering

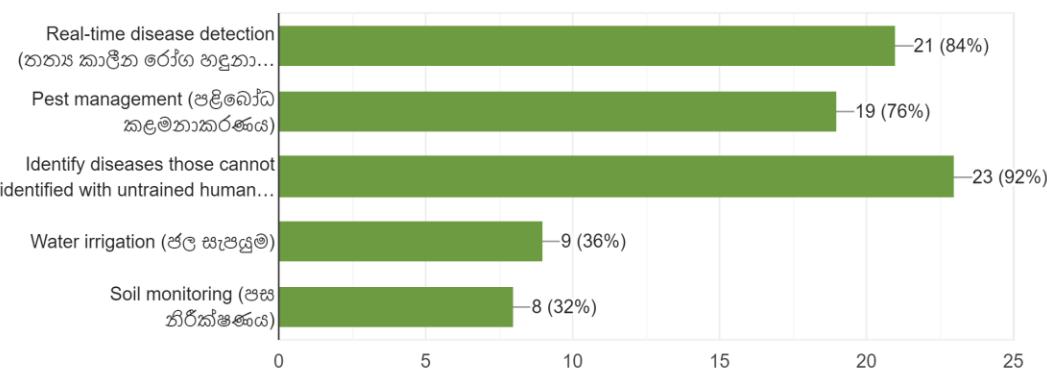
Research Question

Introduction

3. Do you want to increase the yield of the plantation and thus increase the income by identifying the diseases in papaya cultivation and prevent them? (ඒක වැඩිකර ආදායම වැඩිකර ගැනීමට ඔබට අවශ්‍ය ඇති අවස්ථා?)
25 responses



18. Which features would you find most useful in a papaya cultivation monitoring system? (පැපායාල් වගා අධික්ෂණ පද්ධතියක ඔබට වඩාත් ප්‍රයෝග්‍ය වන් ලක්ෂණ මොනවාදා?)
25 responses

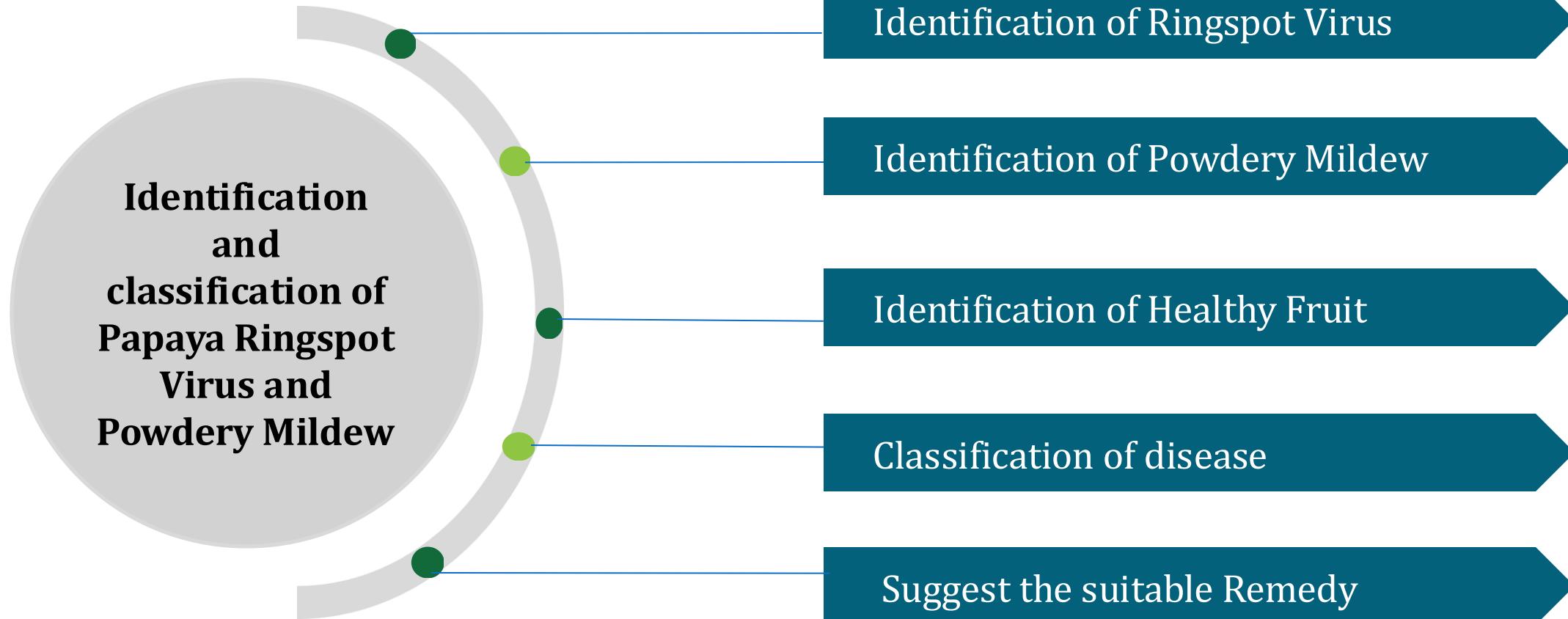


How to identify Papaya Ringspot Virus?

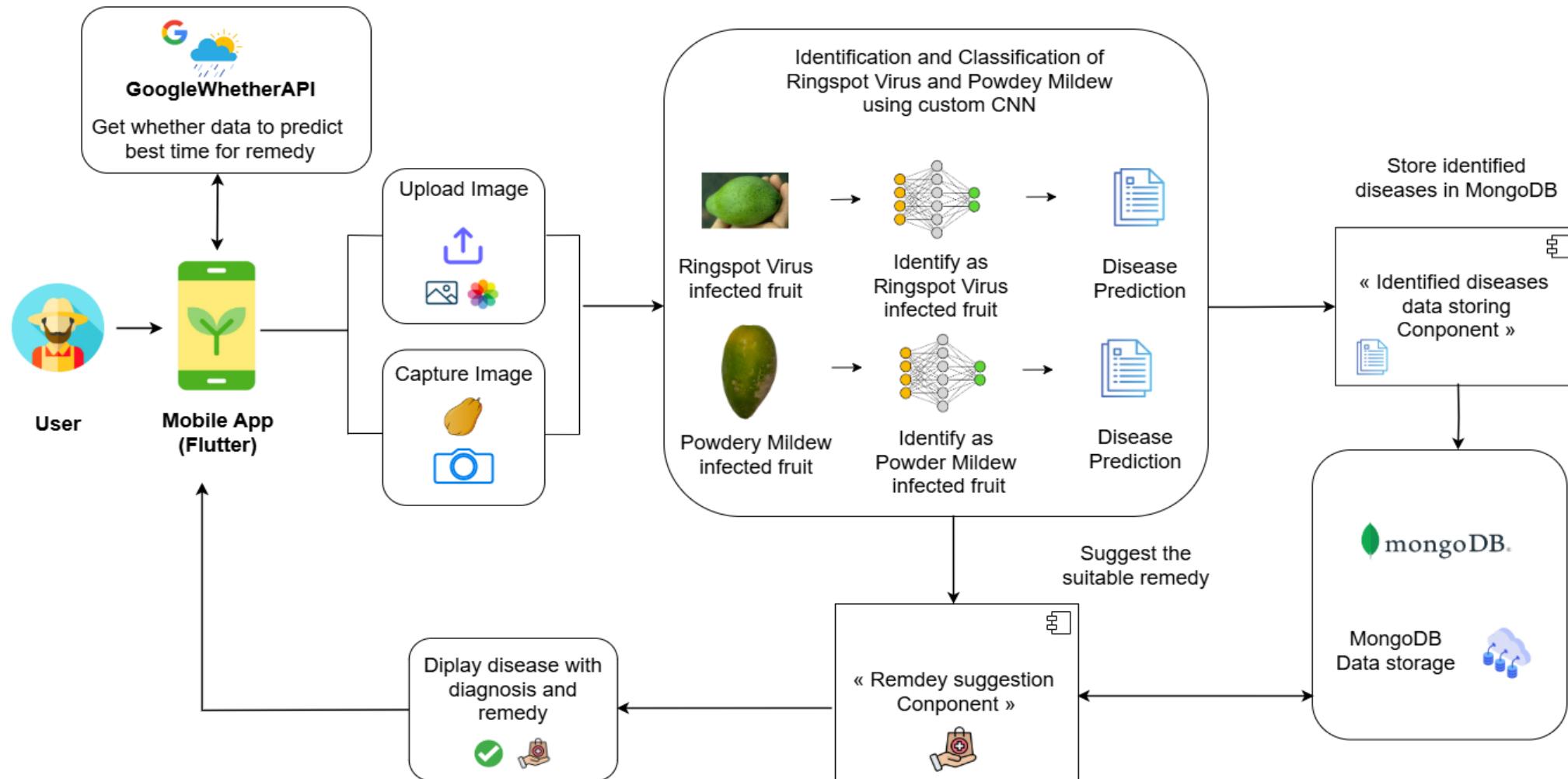
How to identify Powdery Mildew?

Specific and Sub Objectives

Introduction



System Diagram



Evidence of completion



Data Collection



Data pre-processing



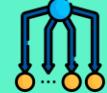
Data visualization



Train YOLOv5 model



Built a custom CNN sequential model



Identify Ring Spot, Powdery Mildew & Healthy Fruit – custom CNN



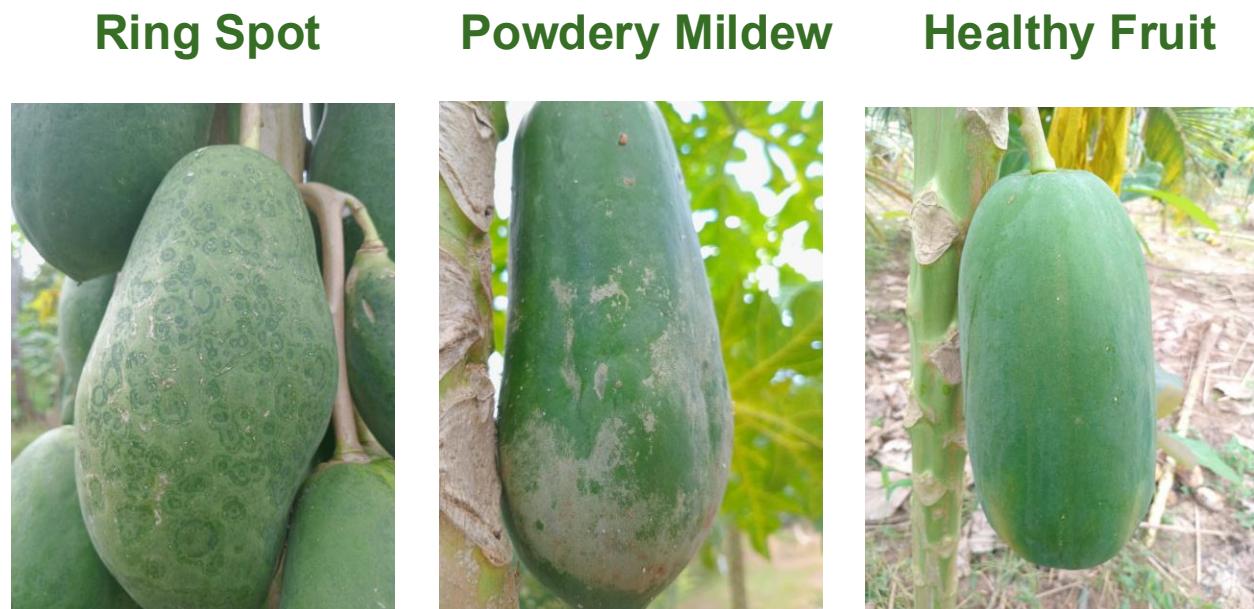
Host the finalized model on Fast API



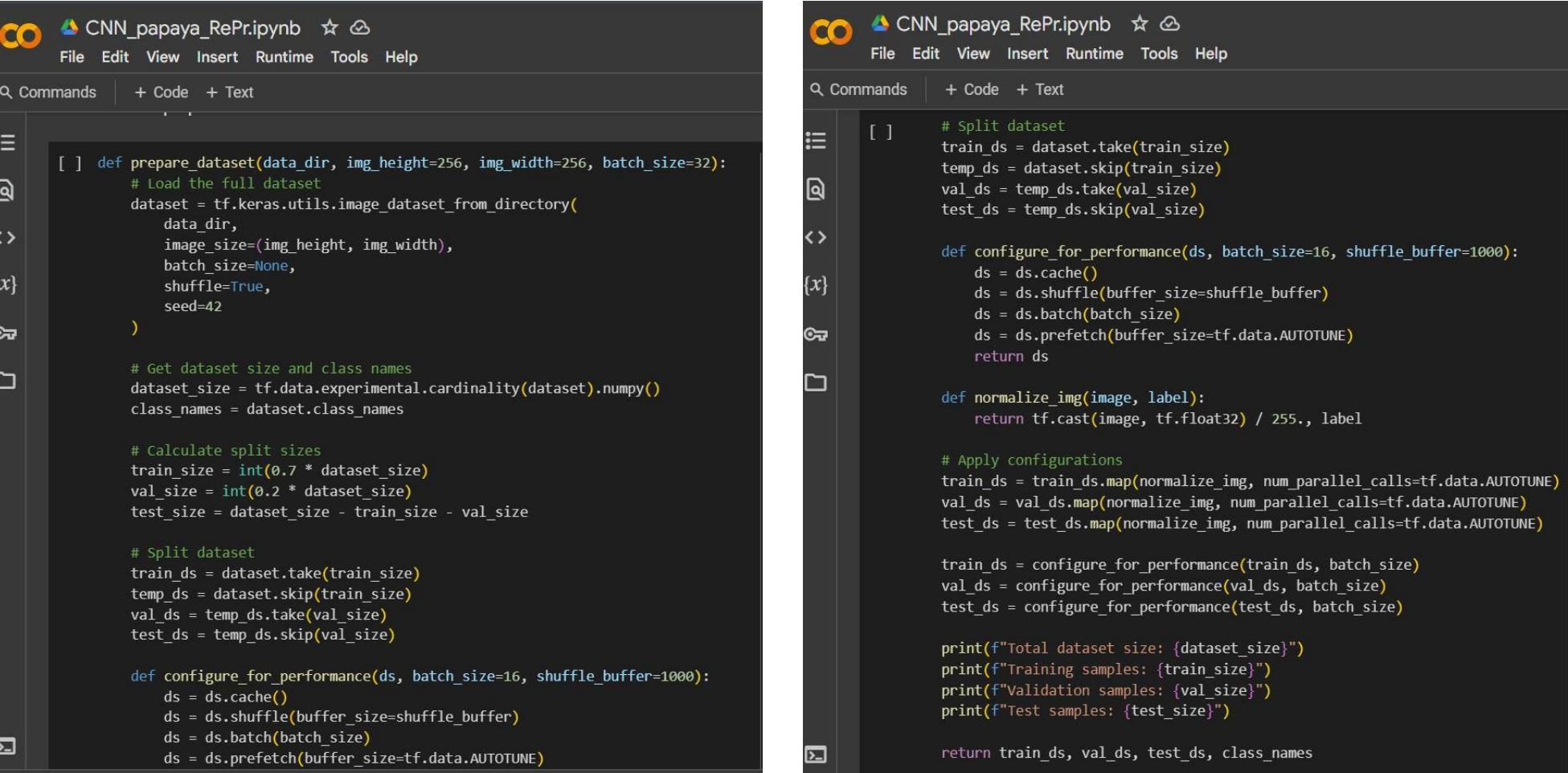
Create the Mobile Application

Data Collection and Pre-Processing

Class Name	Data Set
Ring Spot	500 Images
Powdery Mildew	500 Images
Healthy Fruit	500 Images



Data collection and pre-processing



The image shows two side-by-side screenshots of a Jupyter Notebook interface. Both screenshots have a dark theme and show the same Python script for preparing a dataset.

Code Content:

```
[ ] def prepare_dataset(data_dir, img_height=256, img_width=256, batch_size=32):
    # Load the full dataset
    dataset = tf.keras.utils.image_dataset_from_directory(
        data_dir,
        image_size=(img_height, img_width),
        batch_size=None,
        shuffle=True,
        seed=42
    )

    # Get dataset size and class names
    dataset_size = tf.data.experimental.cardinality(dataset).numpy()
    class_names = dataset.class_names

    # Calculate split sizes
    train_size = int(0.7 * dataset_size)
    val_size = int(0.2 * dataset_size)
    test_size = dataset_size - train_size - val_size

    # Split dataset
    train_ds = dataset.take(train_size)
    temp_ds = dataset.skip(train_size)
    val_ds = temp_ds.take(val_size)
    test_ds = temp_ds.skip(val_size)

    def configure_for_performance(ds, batch_size=16, shuffle_buffer=1000):
        ds = ds.cache()
        ds = ds.shuffle(buffer_size=shuffle_buffer)
        ds = ds.batch(batch_size)
        ds = ds.prefetch(buffer_size=tf.data.AUTOTUNE)
        return ds

    def normalize_img(image, label):
        return tf.cast(image, tf.float32) / 255., label

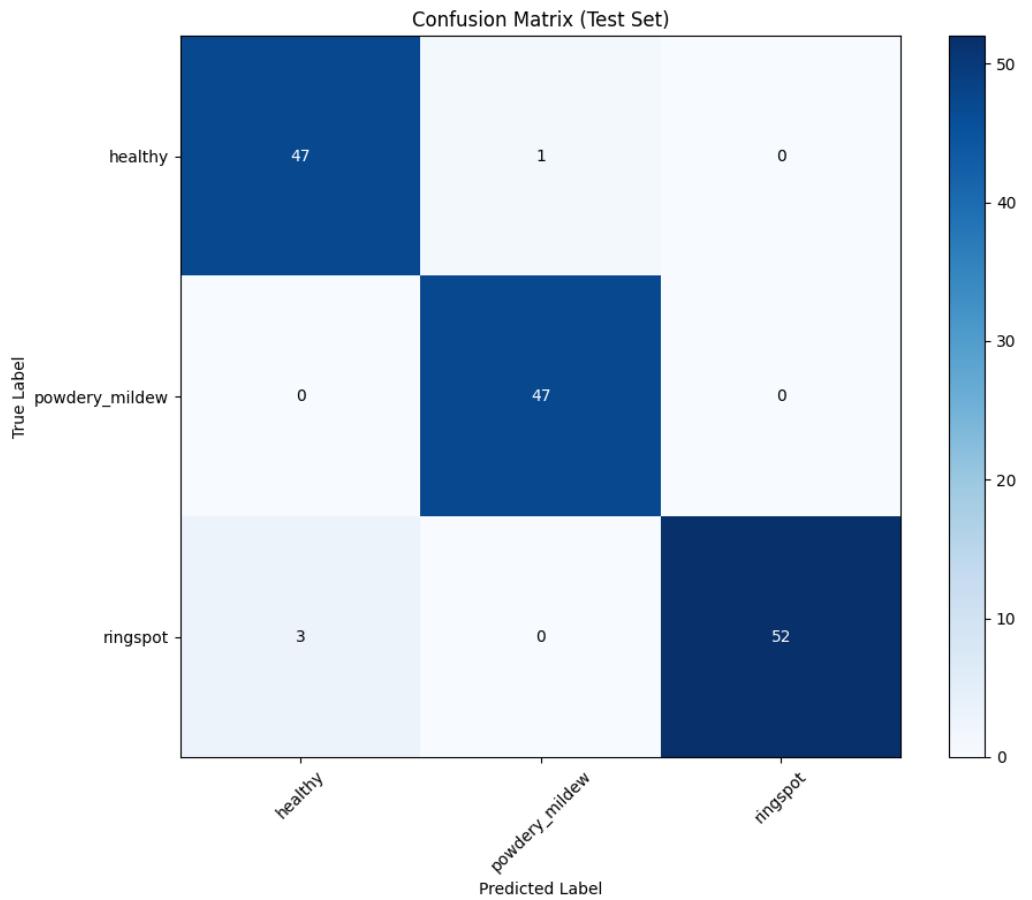
    # Apply configurations
    train_ds = train_ds.map(normalize_img, num_parallel_calls=tf.data.AUTOTUNE)
    val_ds = val_ds.map(normalize_img, num_parallel_calls=tf.data.AUTOTUNE)
    test_ds = test_ds.map(normalize_img, num_parallel_calls=tf.data.AUTOTUNE)

    train_ds = configure_for_performance(train_ds, batch_size)
    val_ds = configure_for_performance(val_ds, batch_size)
    test_ds = configure_for_performance(test_ds, batch_size)

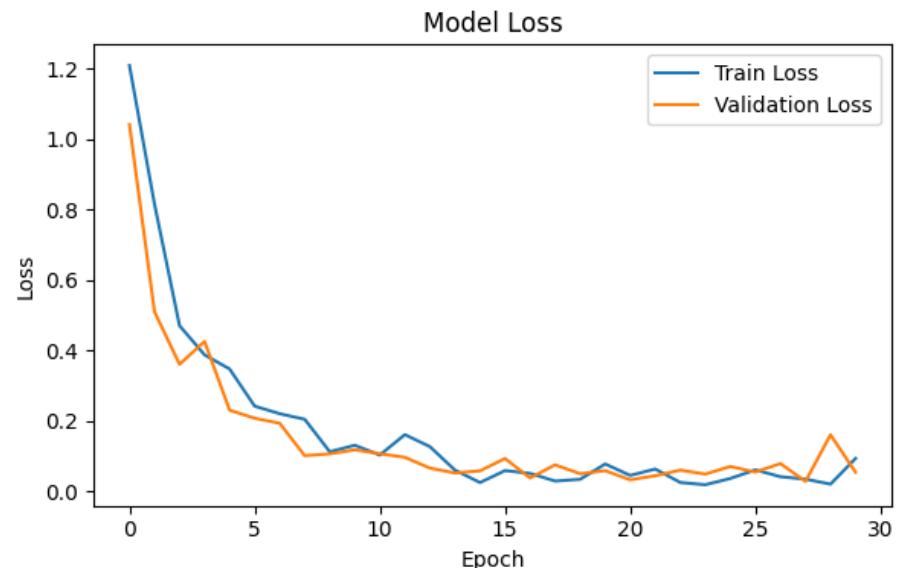
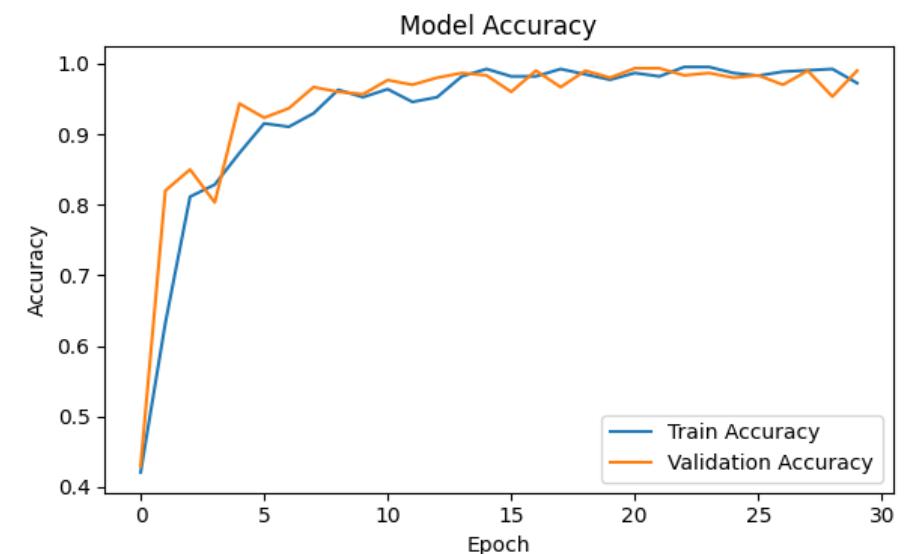
    print(f"Total dataset size: {dataset_size}")
    print(f"Training samples: {train_size}")
    print(f"Validation samples: {val_size}")
    print(f"Test samples: {test_size}")

    return train_ds, val_ds, test_ds, class_names
```

Best performing model



Custom CNN



Model Comparison

Data set Size	1500 Images for custom CNN	1500 Images for YOLOv5 1500 label files(txt)
No of Epochs	30 Epochs used for all classes	
Patience	Patience used as 20 for early stopping	
Image Size	256×256	640×640
No of Parameters	Total params: 8,445,443 (32.22 MB) Trainable params: 8,445,443 (32.22 MB) Non-trainable params: 0 (0.00 B)	Total params: 7,018,216
Model Size	32 MB ~	14 MB ~
Used Memory/ RAM	Tesla T4 GPU	
Used Environment	Google Collab	

Mobile Application Mockups

Evidence of completion

Ring Spot Virus

Viral



1/3 photos

- Ensure proper air circulation by pruning plants, avoid overhead watering to reduce leaf moisture, apply fungicides if necessary, remove infected plant debris, and practice crop rotation to prevent fungal buildup.

Description

Papaya Ringspot Virus (PRSV) is a viral disease affecting papaya plants, leading to severe leaf deformation, mosaic symptoms, and fruit quality degradation.

See more

Symptoms

- Dark Green Rings on fruits
- Uneven yellowing or green mottling on the fruit.
- The skin becomes harder than usual.

See more

Treatment Instructions

Save to Diagnoses

Fruit Disease Detection



No image selected
Take a photo or choose from gallery

Take Photo

Use your camera to capture the maturity

Choose from Gallery

Select an existing photo from your device

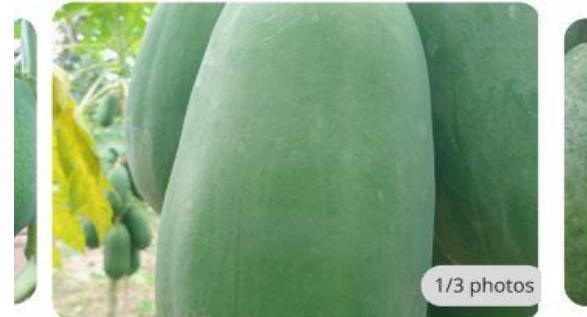
Tips for better detection.

- 1 Ensure good lighting conditions
- 2 Keep the camera steady and focused
- 3 Capture the affected area only.

Predict

Healthy Fruit

Healthy



1/3 photos

- A healthy papaya fruit is free from pest infestations, fungal infections, or physical deformities. It has a uniform shape, bright natural color, and a firm yet slightly soft texture when ripe.

Symptoms

- No visible pests or fungal growth.
- Smooth, blemish-free skin.
- Vibrant color and natural shine.
- Firm texture with no signs of decay.

See more

Save to Diagnoses

Ask Community

Add Image

Improve the probability if receiving the right answer

Add Crop

Your question to the community

Add a question indicating what's wrong with your crop

0/200

Description of your problem

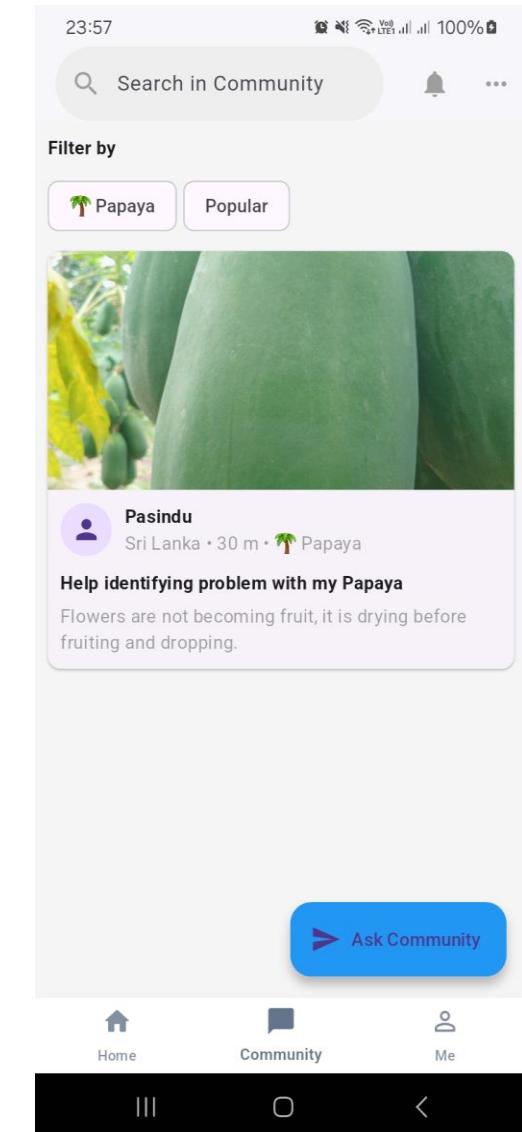
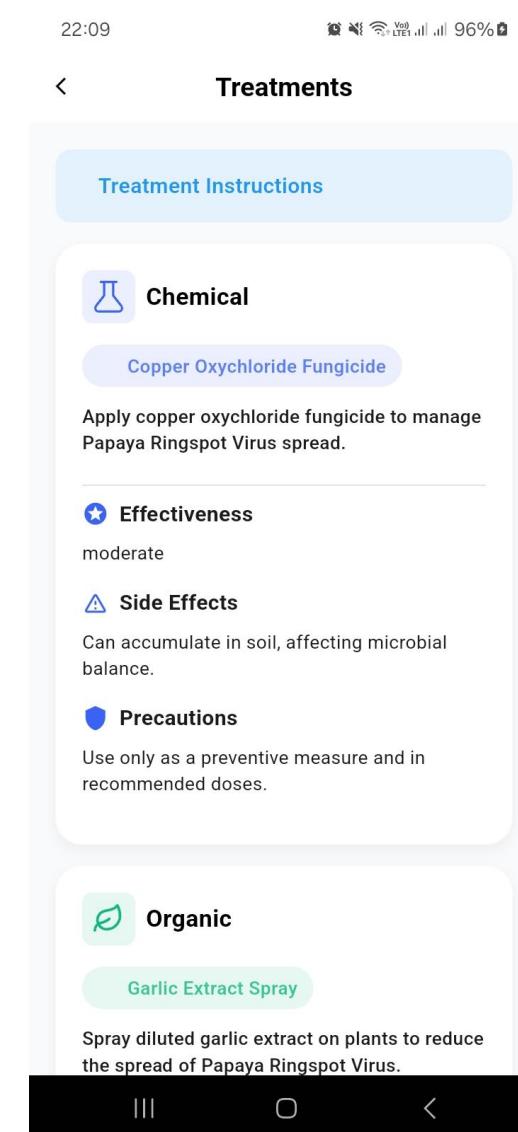
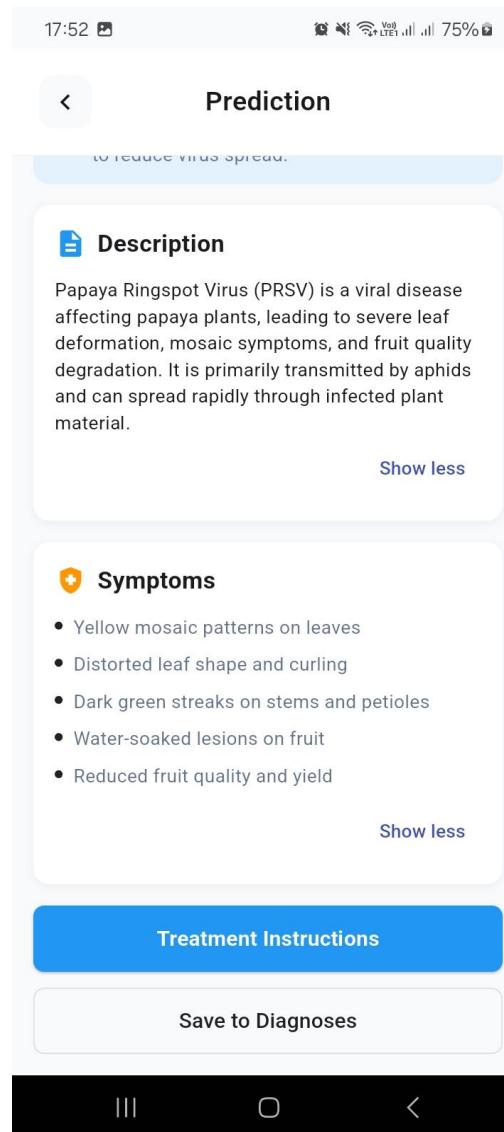
Describe specialities such as change of leaves, fruits, root colour, bugs ...

0/2500

Send

Mobile Application UIs

Evidence of completion





IT21161810 | Mangchanayaka M.V.V

Software Engineering

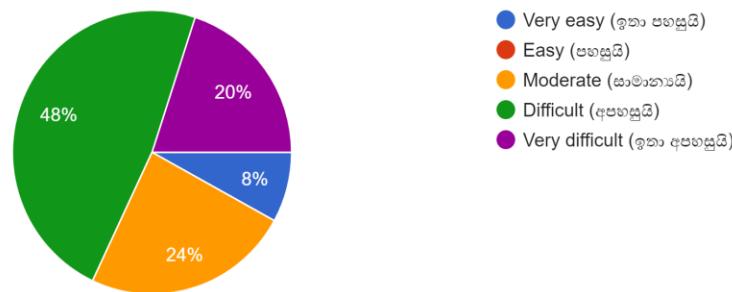
Research Problem

Introduction

- Mite and mealy bug infestations severely damage papaya crops financial losses for farmers [2]
- Early detection and effective management of these pests are essential
- Current systems do not provide personalized and specific remedy suggestions for pest infestations
- Lack of expert knowledge identifying pest diseases
- Existing systems may not be optimized for real-time pest identifications [3]

8. How easy is it to seek domain expertise in your area? (මධ්‍ය ප්‍රදේශයේ පැපාල් සම්බන්ධ විශේෂඥ දෙනුට ලබා ගැනීම කොතරම් පහසුද?)

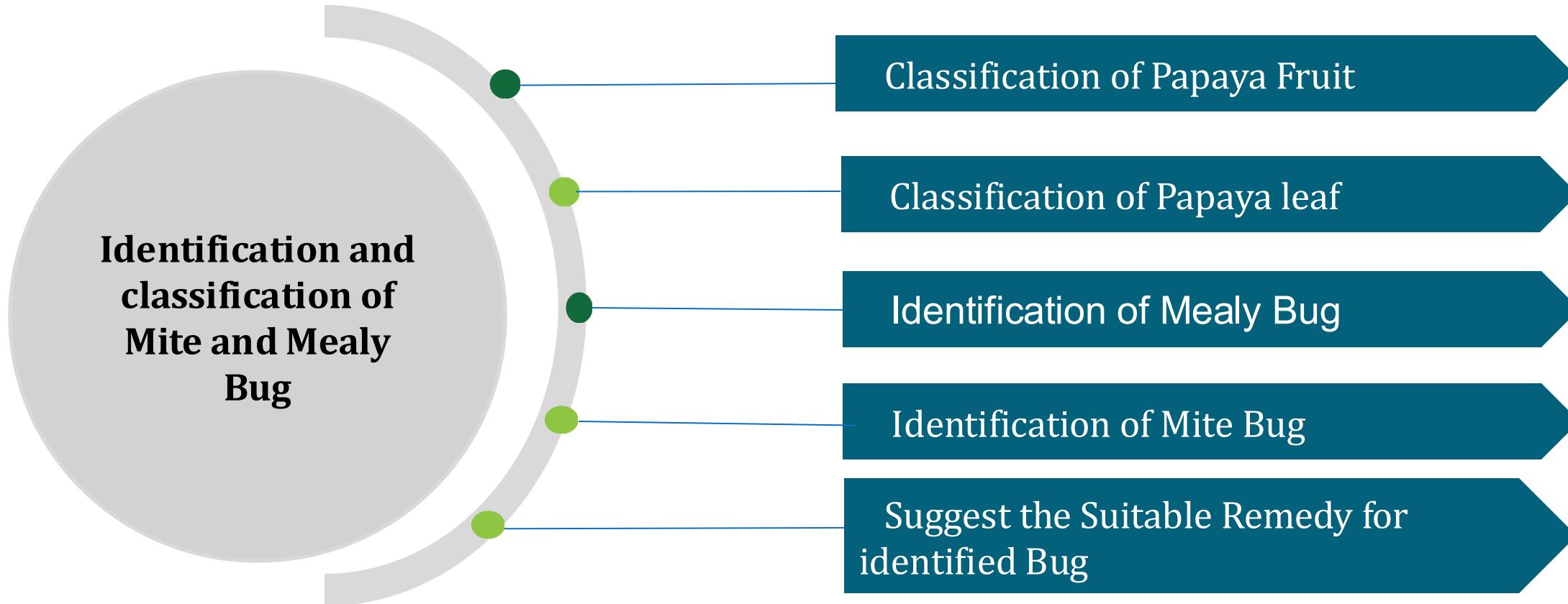
25 responses



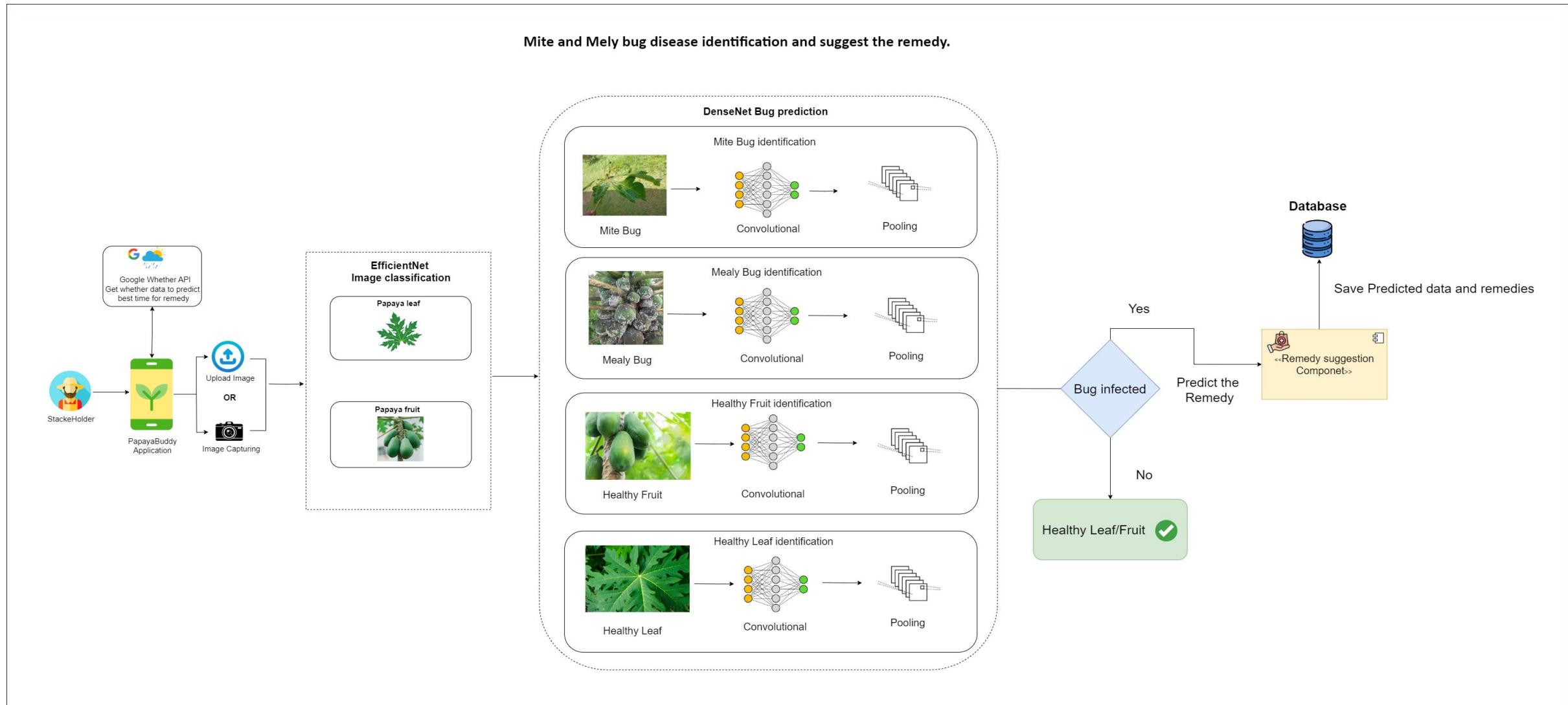
How to identify Mite Bug?

How to identify Mealy Bug?

Specific and Sub Objectives



Methodology System Diagram



Evidence of Completion



Dashboard UI



Pest Image Picker



Node js API Controllers



Disease View & Treatment View



Dense Net CNN for identify Pest infection



Data Preprocessing



Fast API Server to expose CNN model as a endpoint



Efficient Net for healthy leaf & fruit identification

Data Collection

Class Name	Data set
Healthy leaves	310 Images
Healthy Fruit	310 Images
Mealy Bug	300 Images
Mite Bug	300 Images



Healthy Leaf



Healthy Fruit



Mealy Bug



Mite Bug

CNN Model Summary

	DenseNet	EfficientNet
Model Version	DenseNet121(Functional)	EfficientNetv2-B0(Functional)
Parameters	Total : 70,41,604 Trainable params : 41,00	Total : 60,83,538 Trainable params : 60,22,930
No of Classes	4 Healthy Fruit, Healthy Leaf, Mealy Bug, Mite Bug	2 Fruit, Leaf
Dataset Size	600 images	620 images
No of Epochs	30	30
Batch Size	32	16
Input Size	(224 x 224)	(224 x 224)
Used Environment	Jupyter notebook	Jupyter notebook

Data Augmentation & Preprocessing

- Preprocessed to a size of 224 x 224, making it best suited for model training.
- Tensorflow's ImageDataGenerator API for Data Augmentation

```
# Data Generators with Augmentation
train_datagen = ImageDataGenerator(
    preprocessing_function=tf.keras.applications.efficientnet.preprocess_input,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

val_datagen = ImageDataGenerator(preprocessing_function=tf.keras.applications.efficientnet.preprocess_input)
test_datagen = ImageDataGenerator(preprocessing_function=tf.keras.applications.efficientnet.preprocess_input)

train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=image_size,
    batch_size=batch_size,
    class_mode='categorical'
)
```

Data Preprocessing & Augmentation in EfficientNet

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)

val_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow_from_directory(
    'densenet_dataset/splitted_data/train',
    target_size=(224, 224),
    batch_size=32,
    class_mode='categorical'
)
```

Data Preprocessing & Augmentation in DenseNet

Data Visualization for Performance Analysis

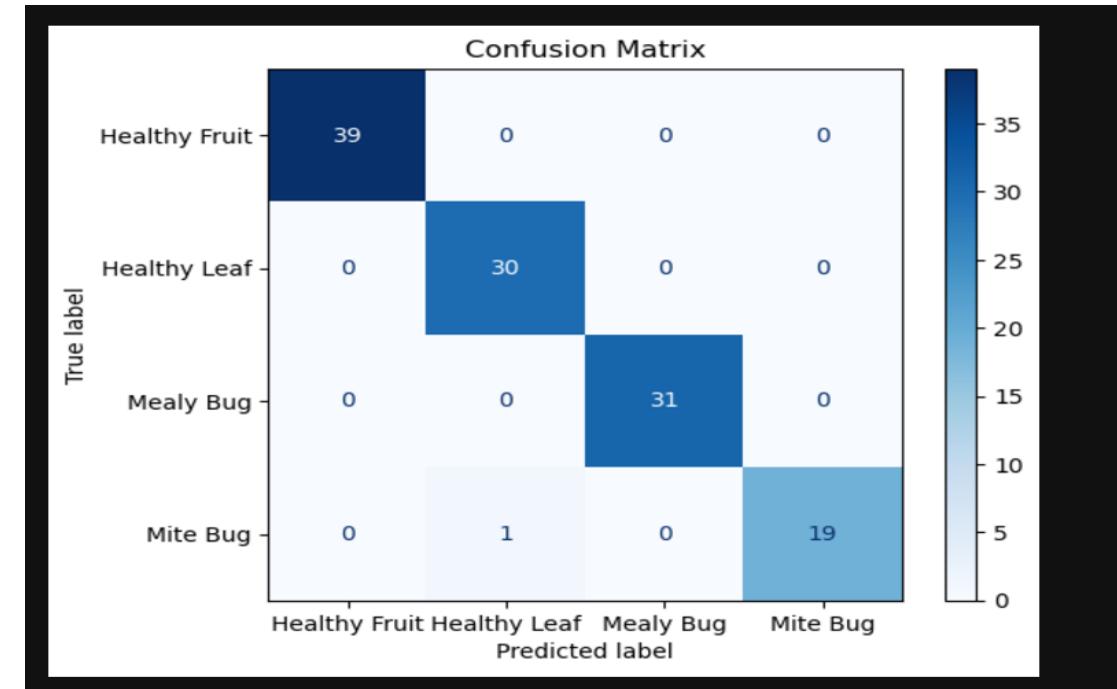
- Confusion Matrix for DenseNet

```
4/4 [=====] - 80s 12s/step - loss: 0.0459 - accuracy: 0.9917
Test accuracy: 0.99
1/1 [=====] - 2s 2s/step
1/1 [=====] - 1s 996ms/step
1/1 [=====] - 1s 1s/step
1/1 [=====] - 1s 756ms/step
Confusion Matrix:
[[39  0  0  0]
 [ 0 30  0  0]
 [ 0  0 31  0]
 [ 0  1  0 19]]

Classification Report:
precision    recall    f1-score   support
Healthy Fruit 1.00 1.00 1.00 39
Healthy Leaf 0.97 1.00 0.98 30
Mealy Bug 1.00 1.00 1.00 31
Mite Bug 1.00 0.95 0.97 20

accuracy 0.99 0.99 0.99 120
macro avg 0.99 0.99 0.99 120
weighted avg 0.99 0.99 0.99 120
```

Predicted label across all true labels



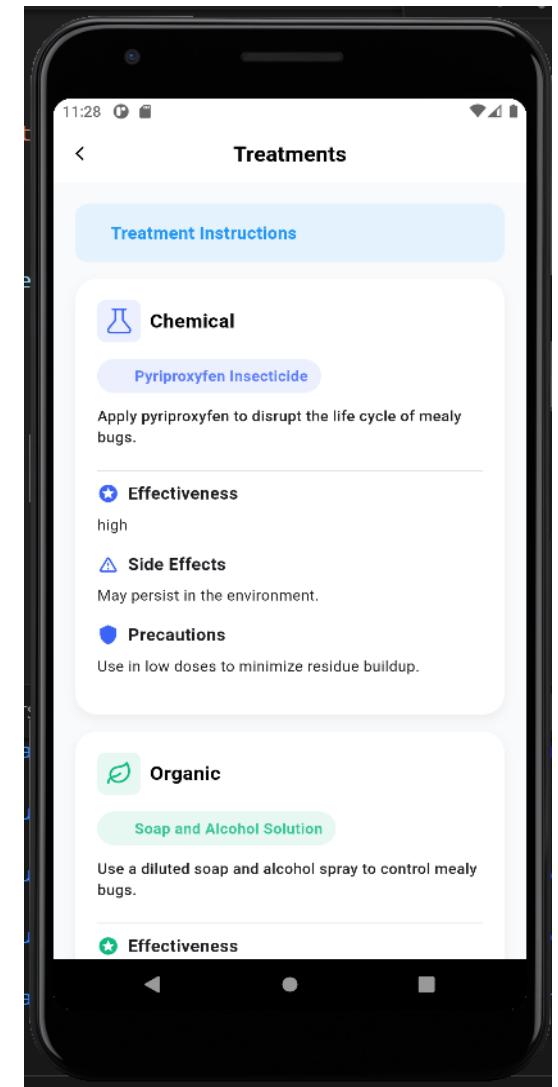
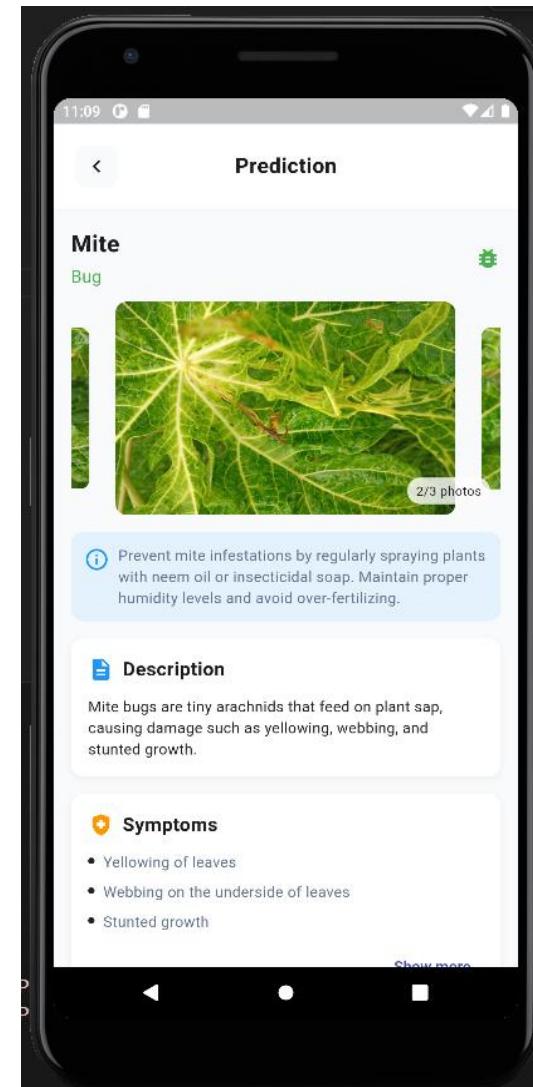
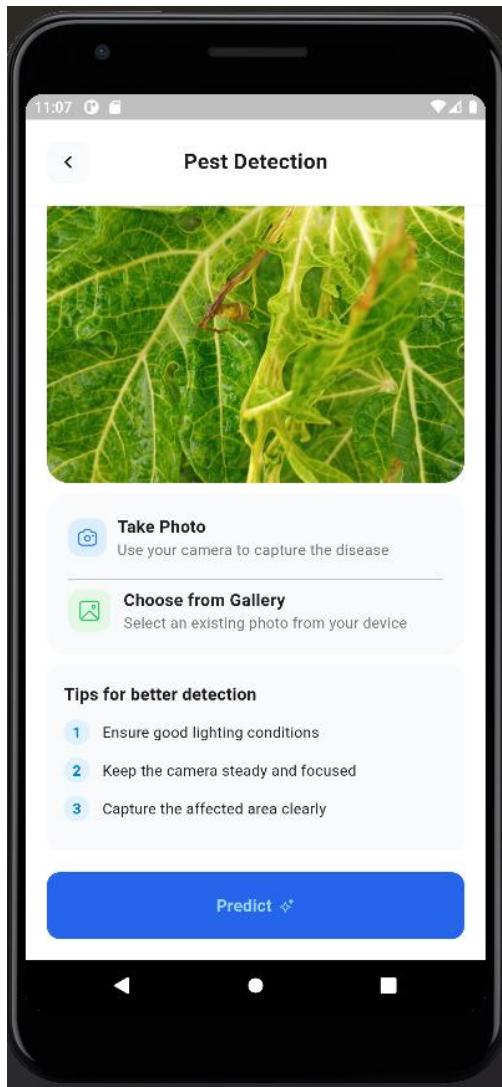
Plotted Predicted label across all true labels

Mobile Application Mockups

The image displays five screens of a mobile application designed for managing plant health and diseases. The screens are:

- Welcome Screen:** Shows a welcome message "Welcome back Alex", the date "Today, Feb 26", and temperature "24°C 25°C". It features a weather icon (sun) and navigation icons.
- Disease Detection Screen:** A central screen for disease detection. It shows a placeholder "No image selected" with instructions to "Take a photo or choose from gallery". Below this are two options: "Take Photo" (use camera) and "Choose from Gallery" (select from device). A "Predict" button is at the bottom.
- Mite Bug Diagnosis Screen:** A detailed diagnosis screen for Mite Bug. It shows a photo of a plant with yellowing leaves. A callout says "1/3 photos". Below the photo are sections for "Treatment Instructions" (prevent mite infestations by spraying neem oil or insecticidal soap), "Description" (mite bugs are tiny arachnids that feed on plant sap, causing damage like yellowing, webbing, and stunted growth), and "Symptoms" (yellowing of leaves, webbing on the underside of leaves, stunted growth, leaf drop). Buttons for "Save to Diagnoses" and "Treatment Instructions" are at the bottom.
- Healthy Fruit Diagnosis Screen:** A diagnosis screen for Healthy Fruit. It shows a photo of a healthy papaya fruit. A callout says "1/3 photos". Below the photo are sections for "Treatment Instructions" (apply imidacloprid to control aphids that spread mosaic virus), "Description" (a healthy papaya fruit is free from pests, fungi, or physical deformities), and "Symptoms" (no visible pests or fungal growth, smooth skin, vibrant color, firm texture). A "Save to Diagnoses" button is at the bottom.
- Treatments Screen:** A list of treatments categorized into "Chemical" and "Organic".
 - Chemical:** Imidacloprid Insecticide. Description: Apply imidacloprid to control aphids that spread mosaic virus. Effectiveness: High. Side Effects: Harmful to pollinators. Precautions: Avoid spraying during flowering periods.
 - Organic:** Pepper Extract Spray. Description: Apply pepper extract to control aphids that spread mosaic virus. Effectiveness: High. Side Effects: Harmful to pollinators. Precautions: Avoid spraying during flowering periods.

Mobile Application UIs





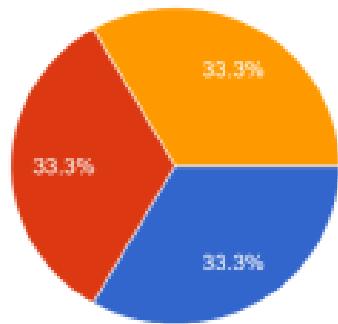
IT21386022 | Senevirathne U.W.H. N

Software Engineering

Research Problem

Introduction

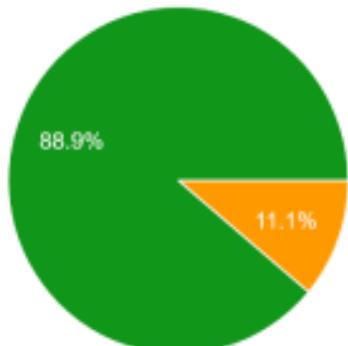
23. How do you currently determine the optimal time for harvesting papaya fruit? (මත දැනට ප්‍රංගයේ සෙවී අභිජන ඇදු යුතුව මාලය තිබු කිරීමෙන් නොවේ?)
9 responses



- Visual inspection (සූචි පිළිබඳ)
- Physical measurements (e.g., size, weight) (සූචි පිළිබඳ (උස් පිළිබඳතා, ප්‍රමාණය))
- Experience-based judgment (සූචි පිළිබඳ ප්‍රතිච්චිතය)

How do you determine the optimal time for harvesting papaya fruit?

25. How significant is the impact of harvesting papayas at the wrong maturity level on your overall yield? (වැරදි පරිගණ මට්ටමකින් ගස්ලටු අභිජනන මිනි සමඟින් අභිජනනට හෙකරු බිඳුනාවටද?)

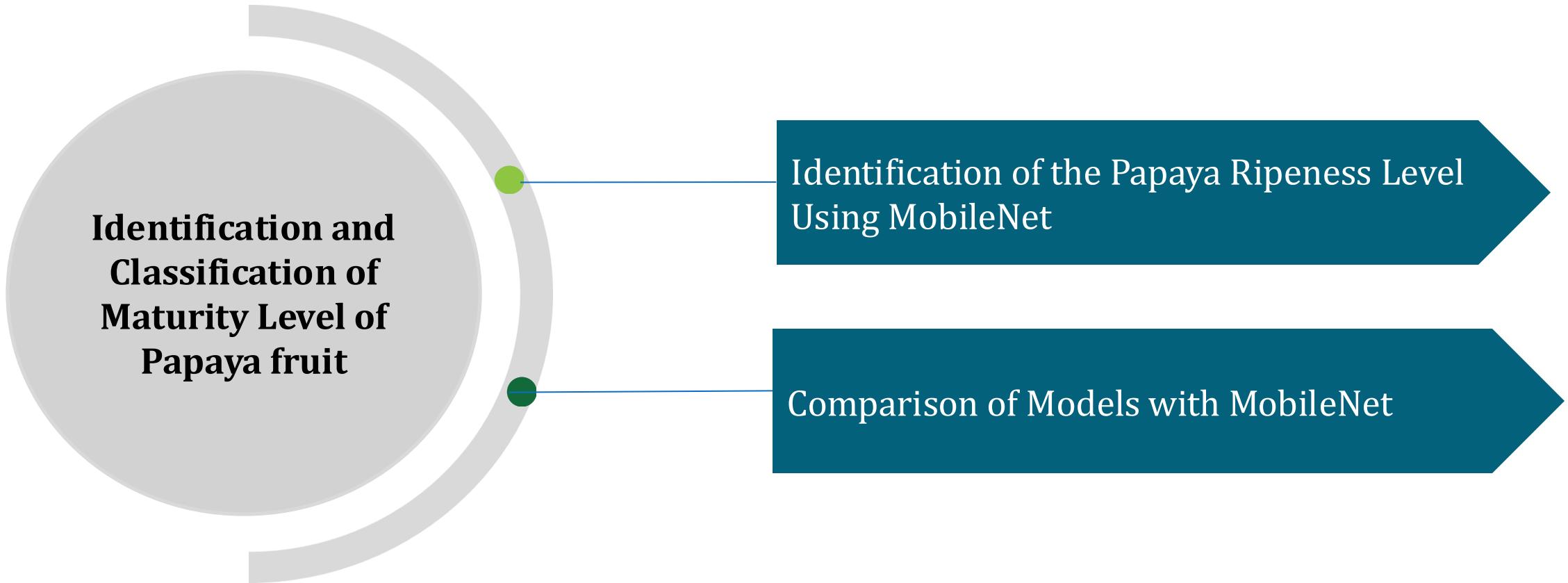


- No impact (බැඳුනාව නැත)
- Minor impact (පැහැ බැඳුනාවේ)
- Moderate impact (යාම්පා බැඳුනාවේ)
- Major impact (බැඳු බැඳුනාවේ)

How significant is the impact of harvesting papayas at the wrong maturity level on your overall yield?

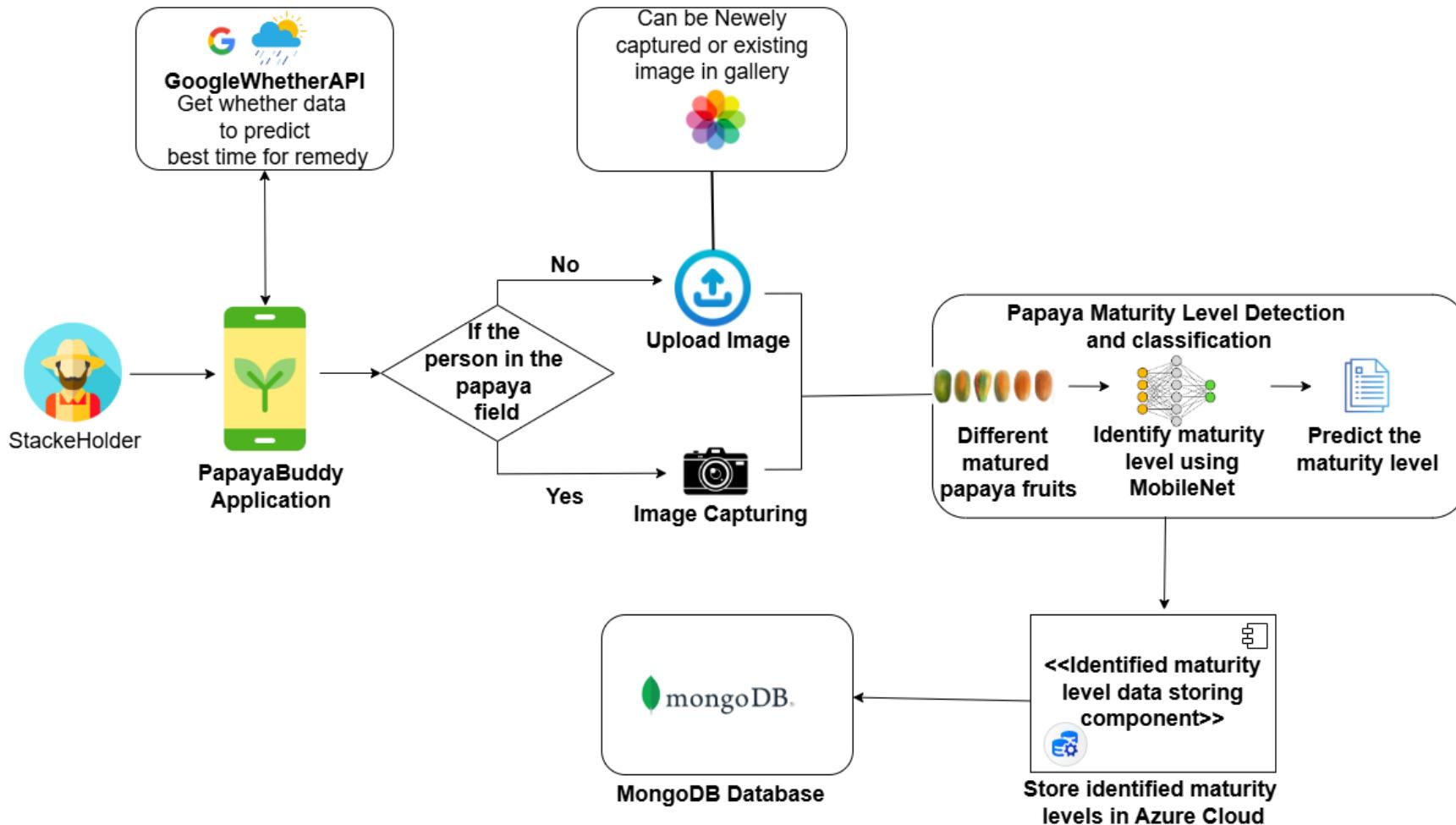
Specific and Sub Objectives

Methodology



System Diagram

Papaya Maturity Level Detection



Evidence of completion



Maturity Piker UI



Maturity Stage UI



Maturity Prediction Node Controllers



Implement MobileNetV2 Model



Host the finalized model in the flask server

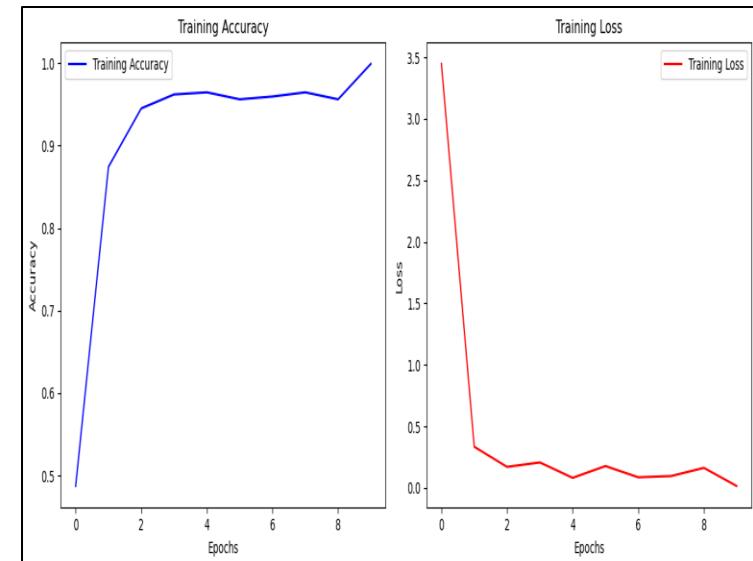
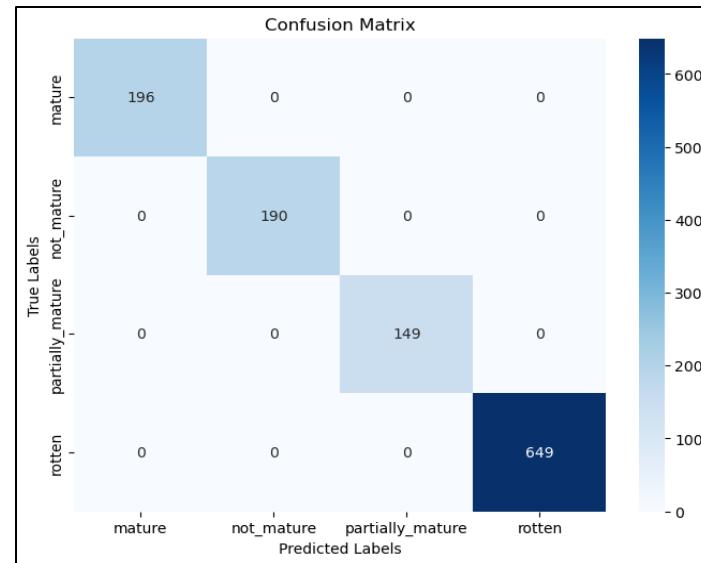


Display the results in mobile application

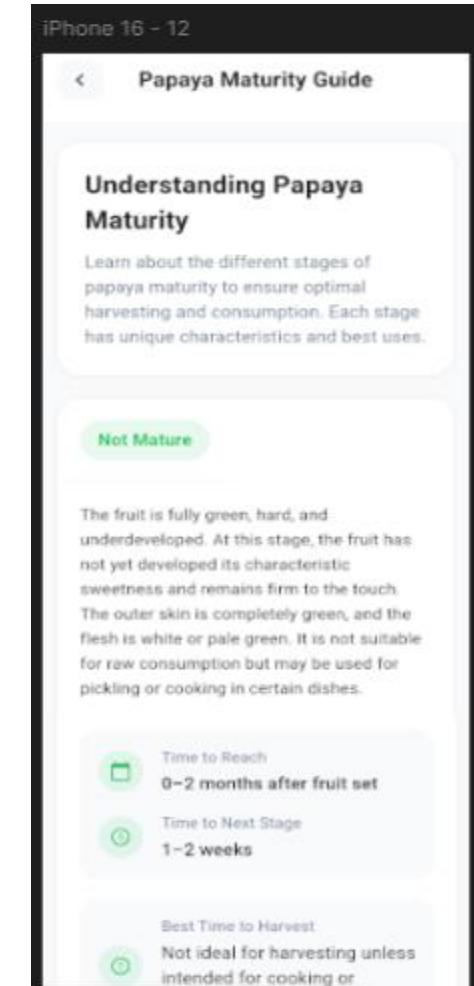
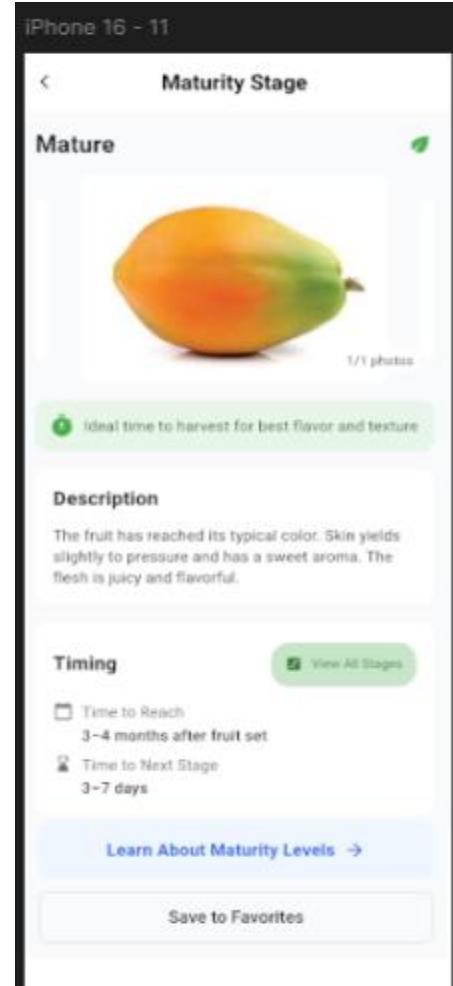
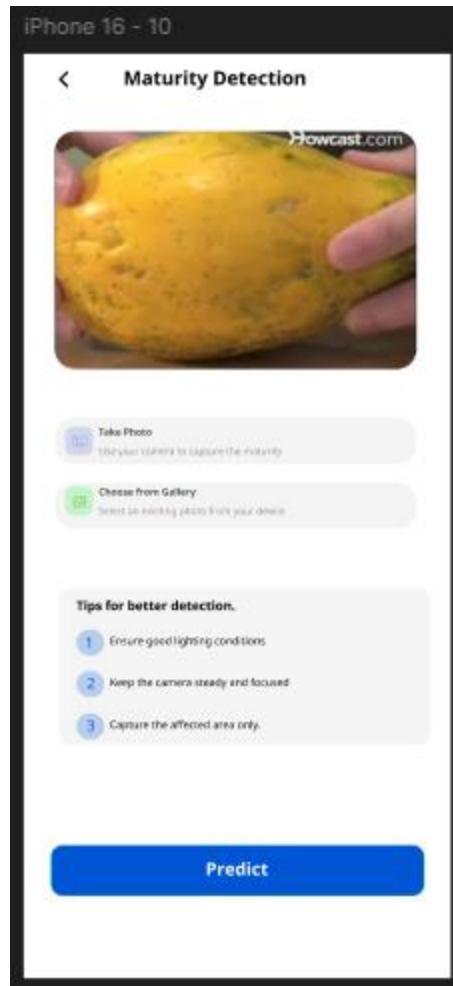
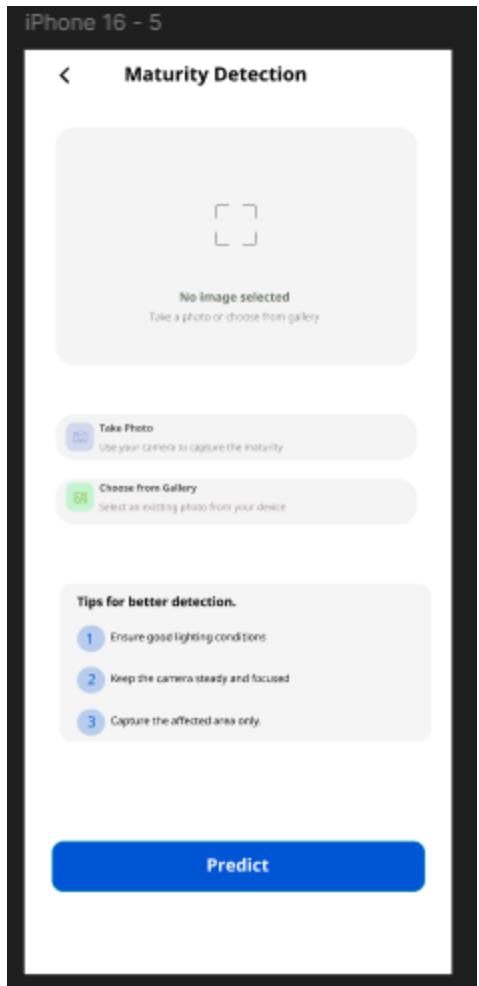
MobileNetV2 Model Evaluation

Evidence of Completion

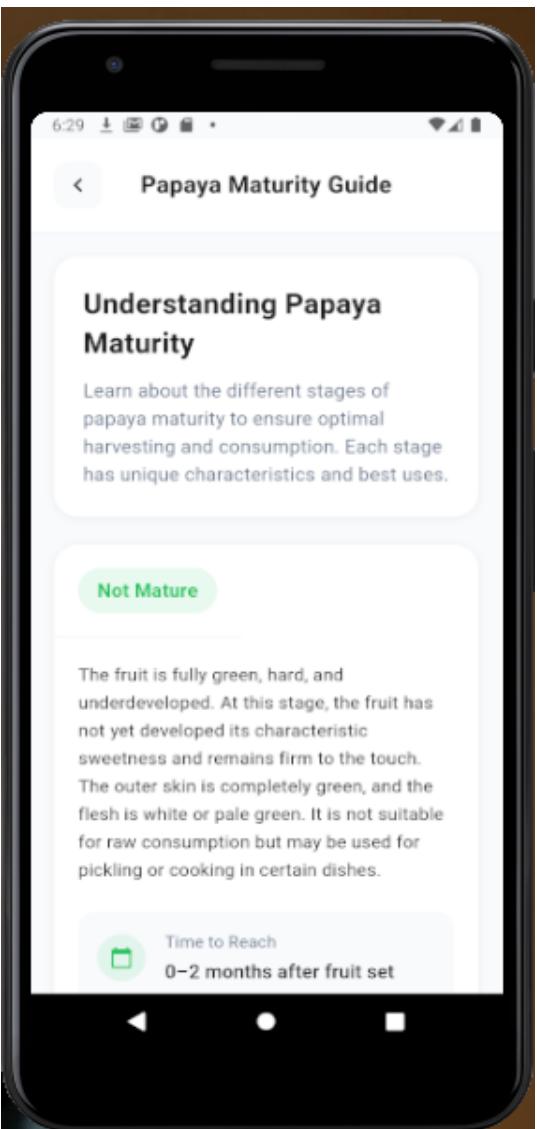
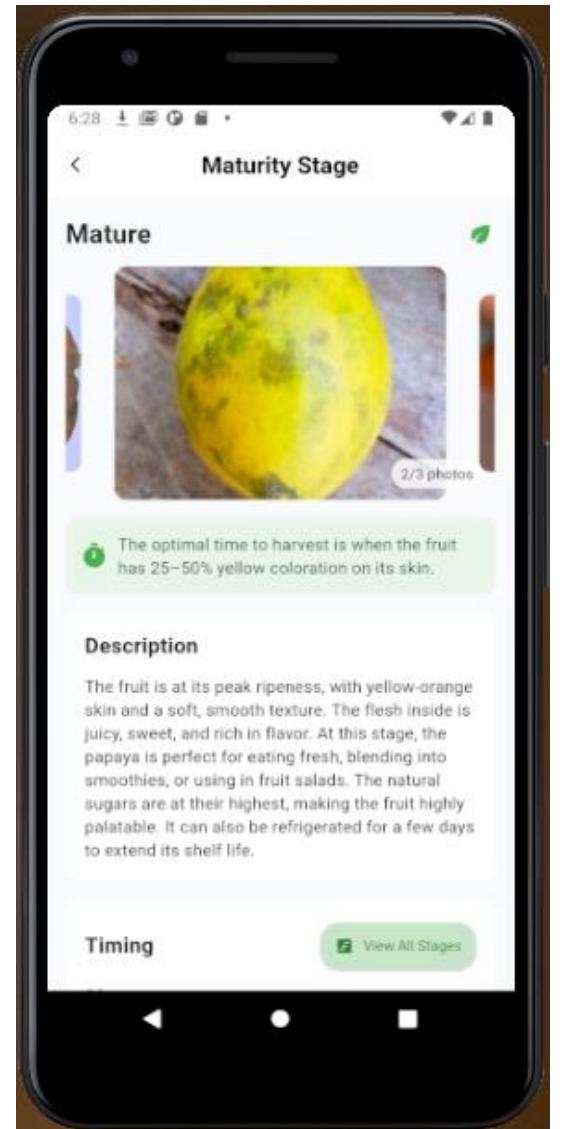
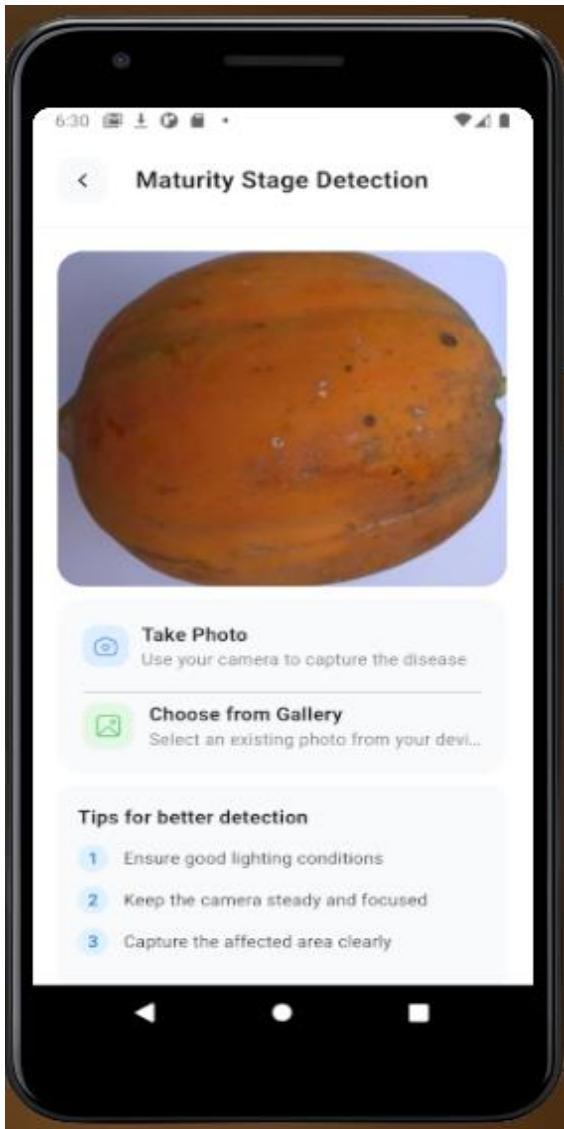
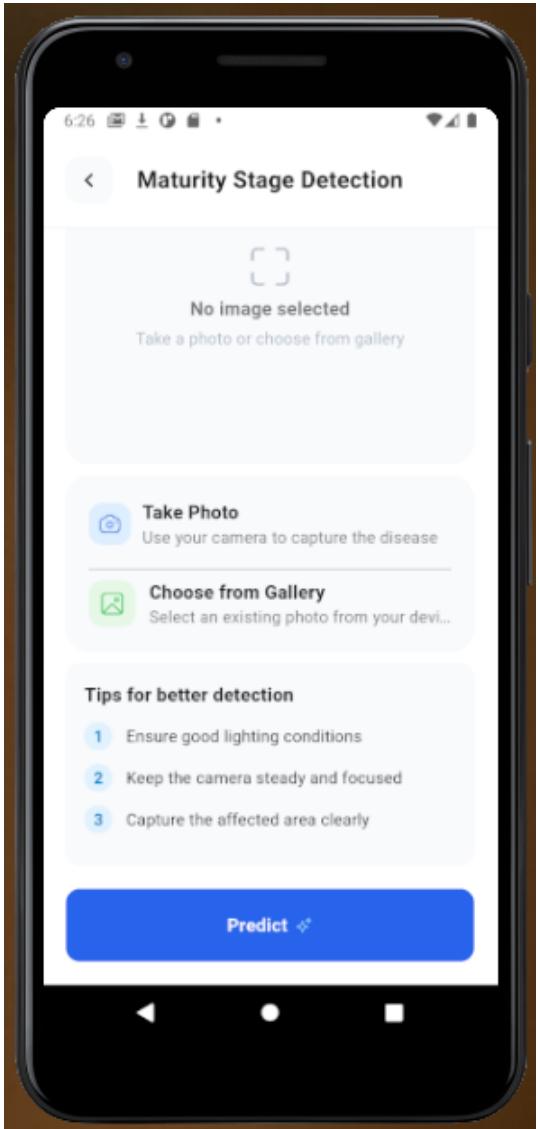
Variables	MobileNetV2
Modal Version	MobileNet2
No of Classes	4
Dataset Size	2000 images
No of Epochs	10
Batch Size	32
Input Size	(224 x 224)
Used Environment	Jupyter notebook

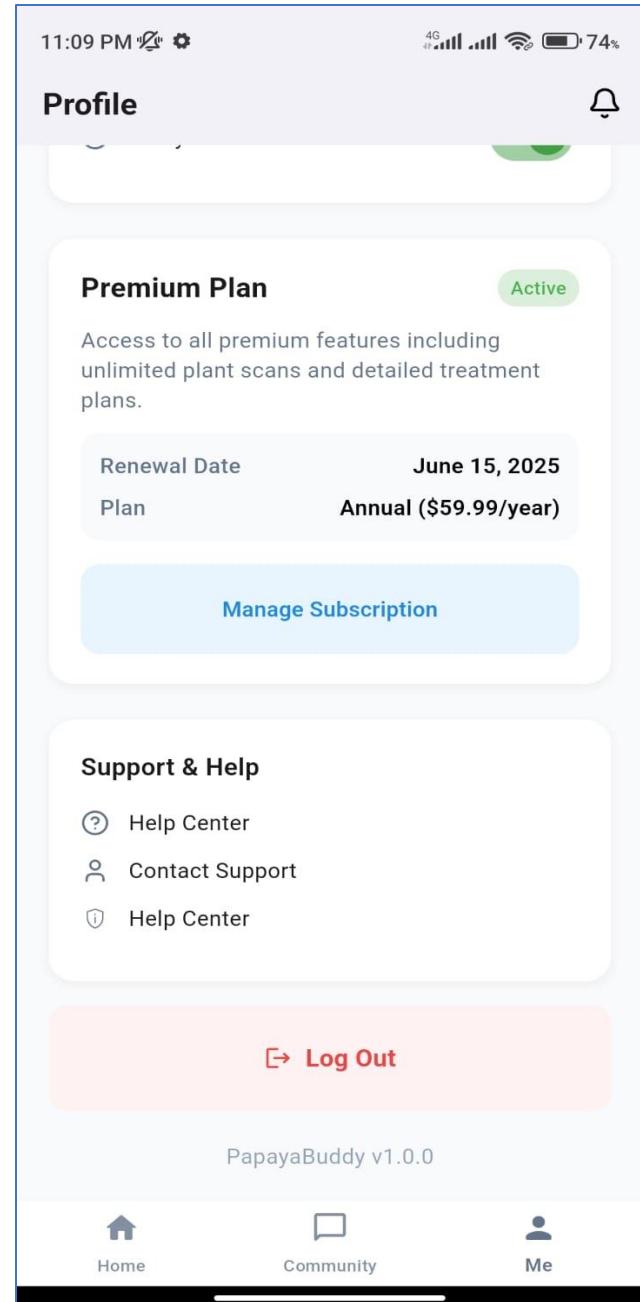


Mobile Application Mockups



Mobile Application UIs





Commercialization

Commodity version

Identification of,

- Cercospora
- Mosaic Virus
- Powdery Mildew
- Ring Spot
- Identification of Bugs
- Fruit Maturity
- Remedy Suggestion

Up to 15 images/ free week

Premium version

- **Commodity version +**
- **With the lowest price**

Up to 100 Images

- Monthly - Price: Rs. 2500.00
- Unlimited Images
- Yearly - Price: Rs. 10000.00

Target Audience

- Farmers
- Stakeholders
- Researchers

Market Place

- Excellent user experience through simpler User Interface
- No need of advanced technical knowledge

Completion and Future work

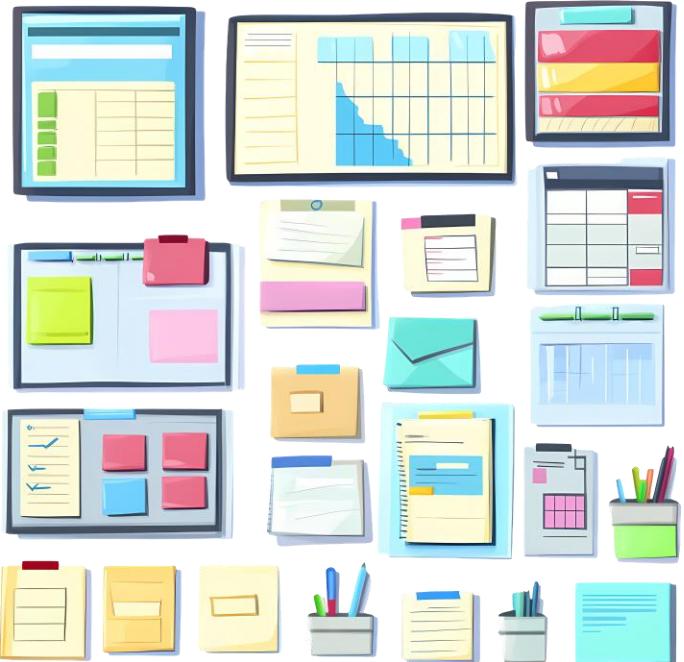
Completion of the Components

- Fruit/ Leaf/ Pest disease detection
- Maturity level detection
- User Management
- Remedy Suggestion
- Community Component View
- Integrate Whether API
- Fast API/Flask API integration with Models



Future implementation

- Develop Community Functionality
- Deploy PapayaBuddy app in Play-Store



Key Challenges & Solutions

- Containerizing services with Docker in the Microservices architecture, including inter-service communication, performance optimization, and network configurations.
- Encountered dependency version conflicts with libraries like TensorFlow, NumPy and etc. To mitigate this, we implemented a Microservices architecture with separate models, ensuring isolation and compatibility.
- Monolithic architecture required excessive computing resources, leading to limitations. To overcome this, we adopted a Microservices architecture for better resource efficiency and scalability.
- Data collection required long-distance travel with high costs, and we faced challenges such as bad weather conditions and poor lighting, affecting data quality.
- Poor camera setup impacted image quality, and the costs for Colab and Google Drive storage added to the challenges.
- Implemented a Microservices architecture to resolve dependency conflicts, with Node.js for data handling, FastAPI for model hosting, and Kong as an API gateway for authentication, authorization, load balancing, rate limiting, and secure service routing.





Papaya Buddy

Thank You !