Detection and Prevention of Citrus Diseases using Deep Learning

Team Members:
Prahlad Acharya (THA077BEI031)
Rijan Pokhrel (THA077BEI034)
Sakar Dahal (THA077BEI037)
Vishal Sigdel (THA077BEI048)

Under the supervision of:
Er. Ritu Raj Lamsal

Department of Electronics and Computer Engineering
Institute of Engineering, Thapathali Campus

July 21, 2024

Presentation Outline

- Methodology
- Results
- Discussions

Methodology

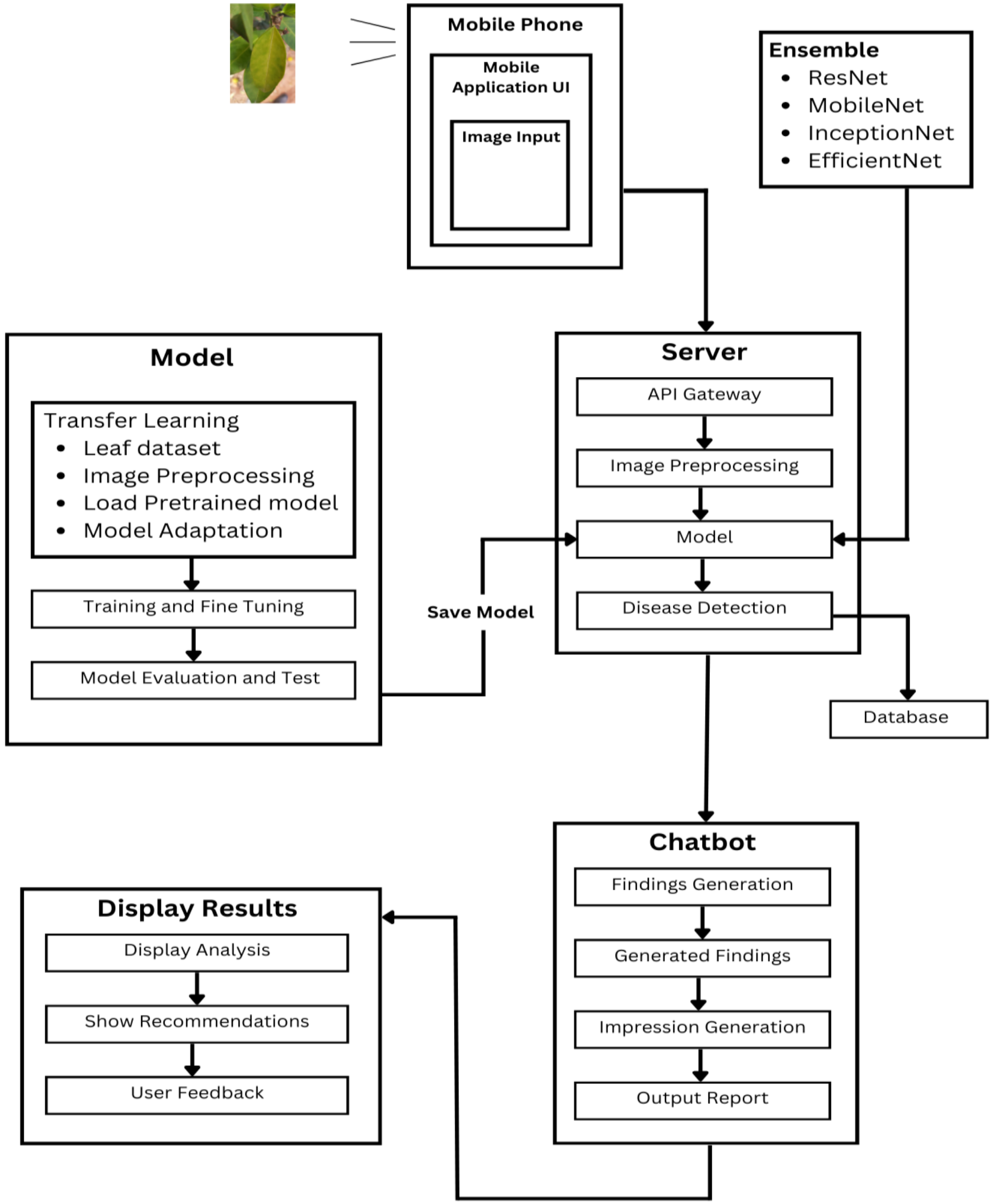
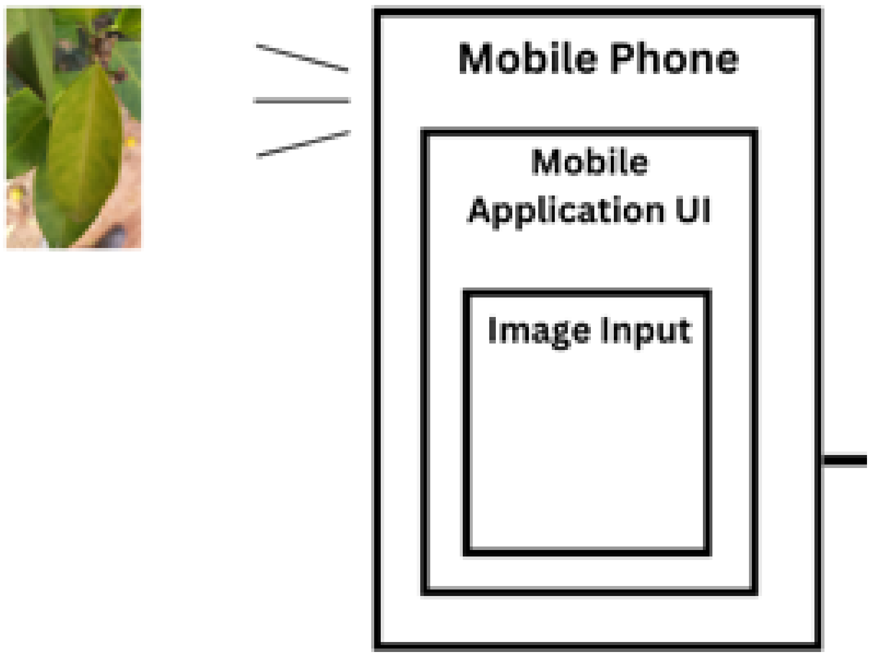


Fig: Block Diagram of overall Project

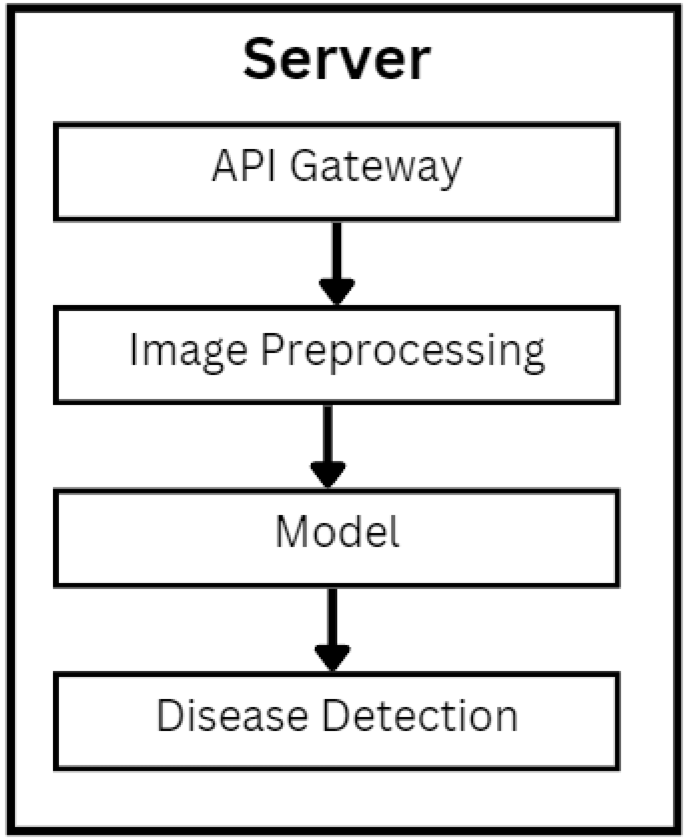
Methodology [2]
Mobile Phone

- Mobile Application is developed using Flutter
- Mobile Application has feature of capturing image
- Image output is fed into the system



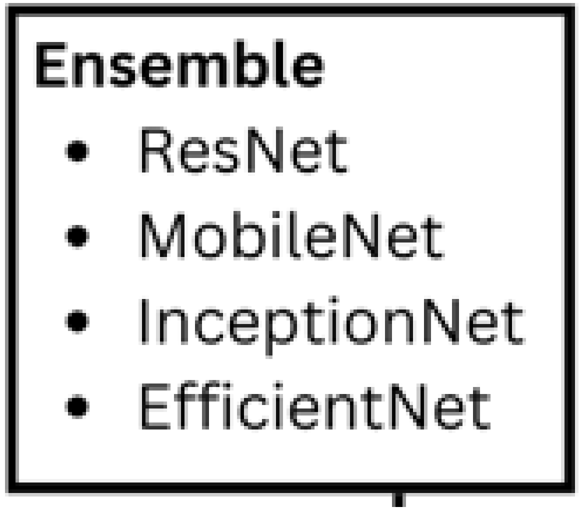
Methodology [2]
Server

- Server is developed using API Gateway
- Input image is pre-processed to meet input requirements
- The pre-processed image is fed into the model
- The model will analyze the input and give predictions



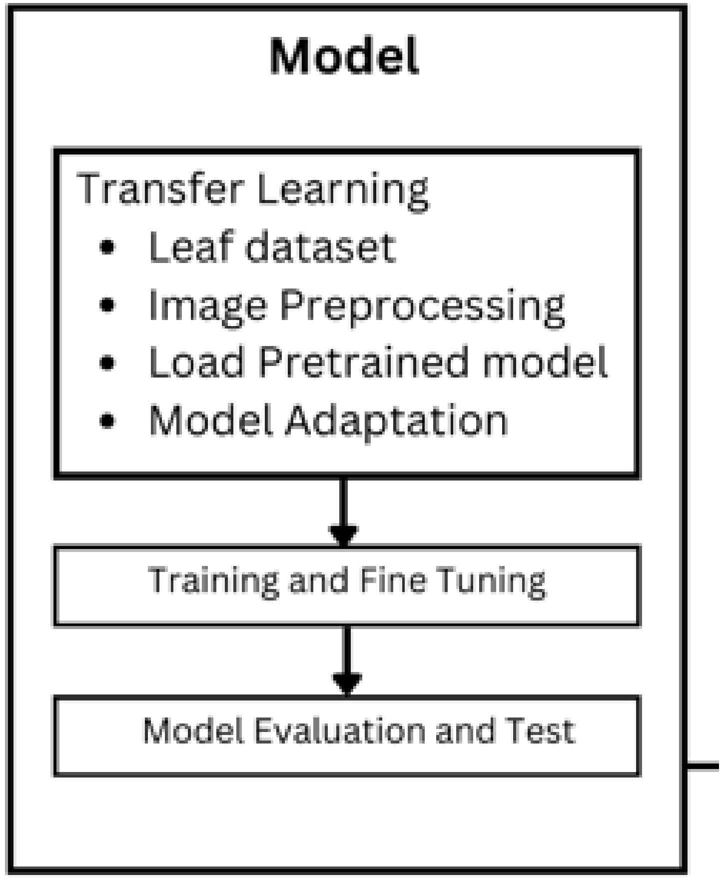
Methodology [4]
Ensemble

- The model is developed by ensembling multiple pre-trained models.
- Ensemble method improves accuracy, robustness and diversity
- Currently discussed pre-trained models are: ResNet, MobileNet, InceptionNet and EfficientNet



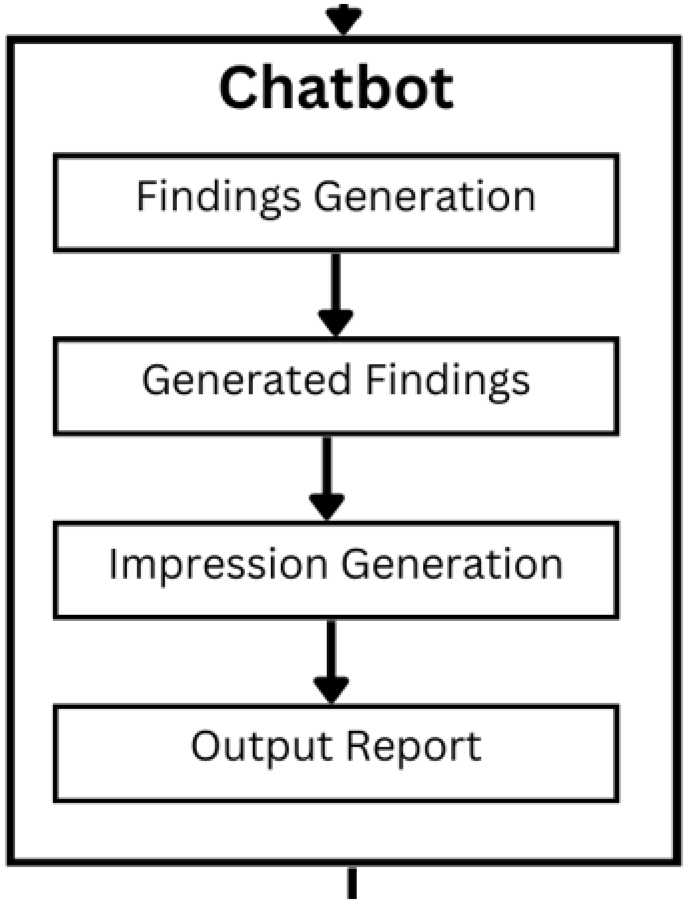
Methodology [4]
Model

- Model uses transfer learning to adapt pre-trained model
- Leaf dataset is pre-processed and used for adaptation
- Model parameters are adjusted to improve performance
- Test and Evaluation for assessing accuracy



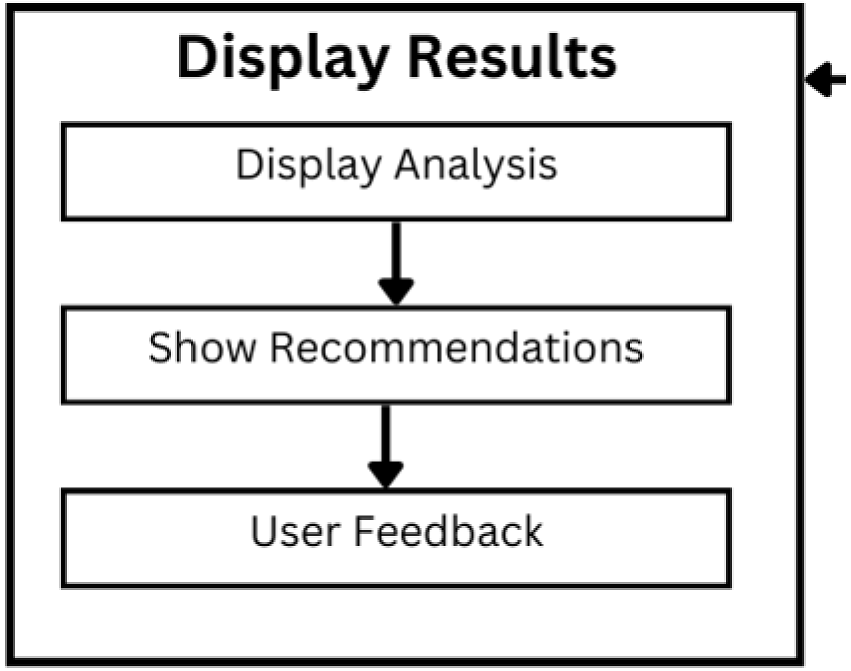
Methodology [5]
Chatbot

- Model’s Prediction is used to create a summary of finding.
- The generated finding is analyzed
- Conclusion is formed using the analysis
- The output is provided to the user

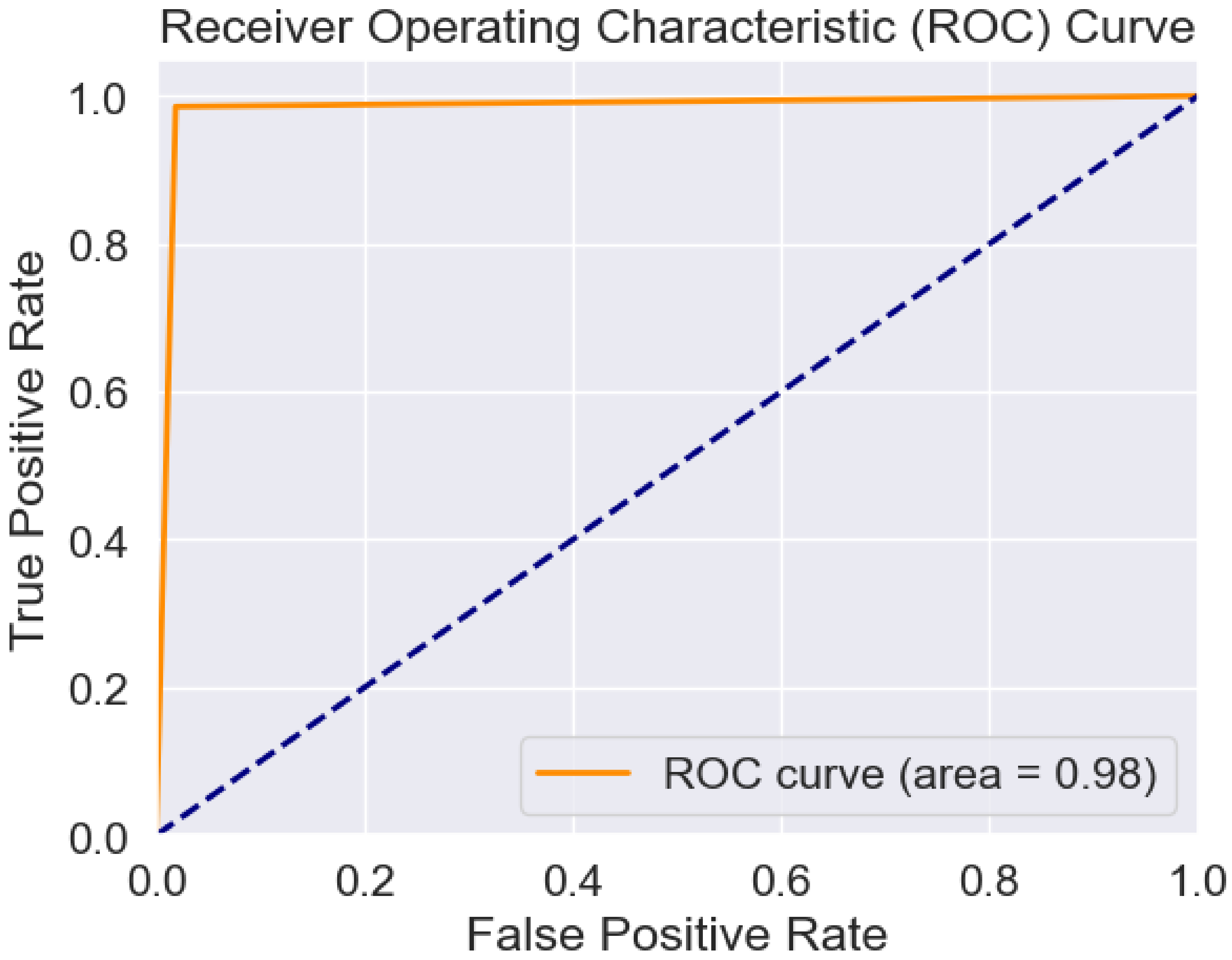


Methodology [6]
Display Results

- Output is presented to user in an easy-to-understand format.
- Disease management techniques are suggested
- User's feedback is taken
- Feedback is used for system improvement



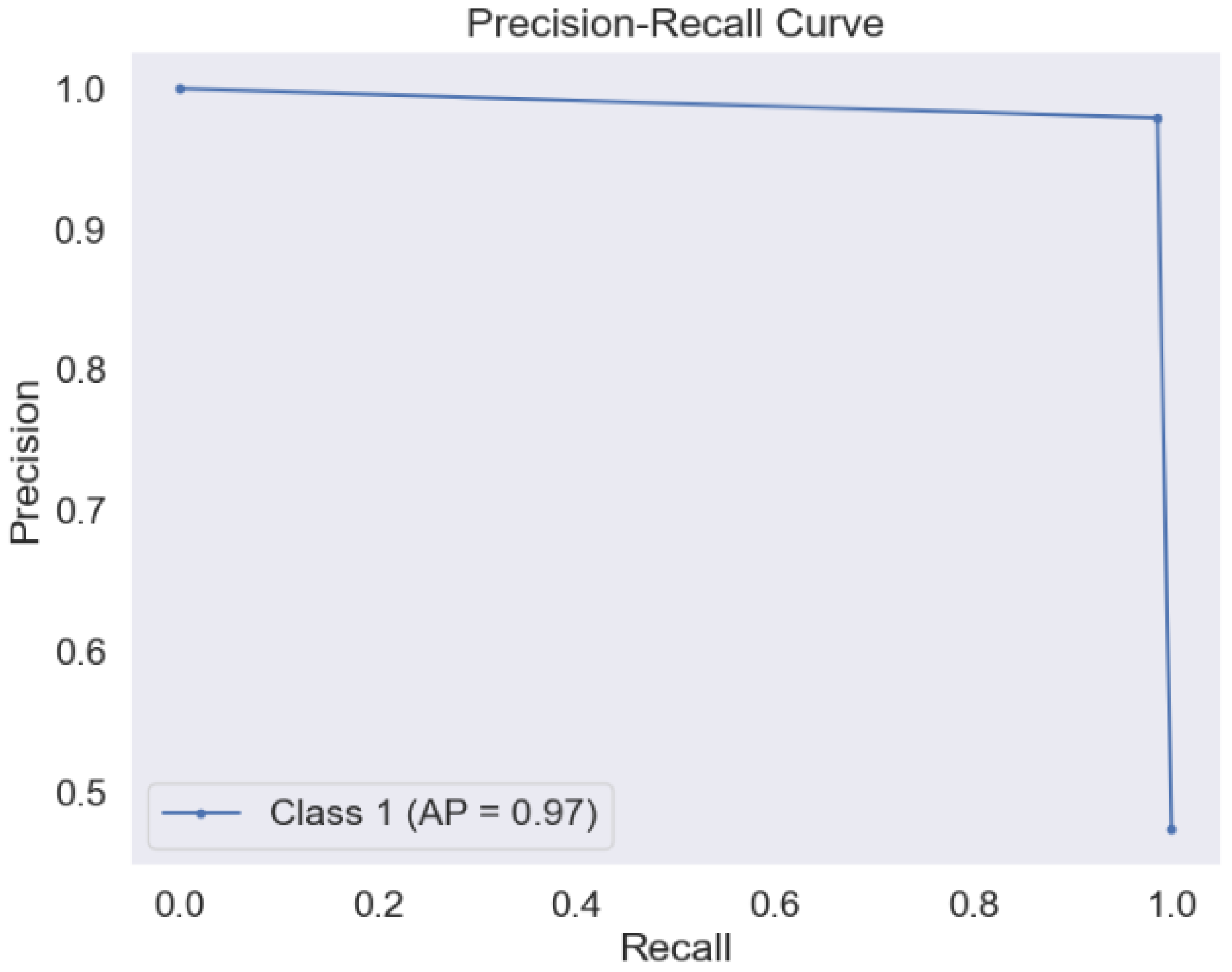
Result: MobileNet [1]
ROC Curve



Result: MobileNet [2]
ROC CURVE

- Area Under the Curve (AUC) is 0.98, indicating near-perfect classification.
- The ROC curve shows exceptional performance of the CNN model in predicting HLB disease.
- The curve rises sharply to a true positive rate of almost 1.0 at a very low false positive rate.
- The model demonstrates high sensitivity and specificity in distinguishing between healthy and diseased citrus plants.

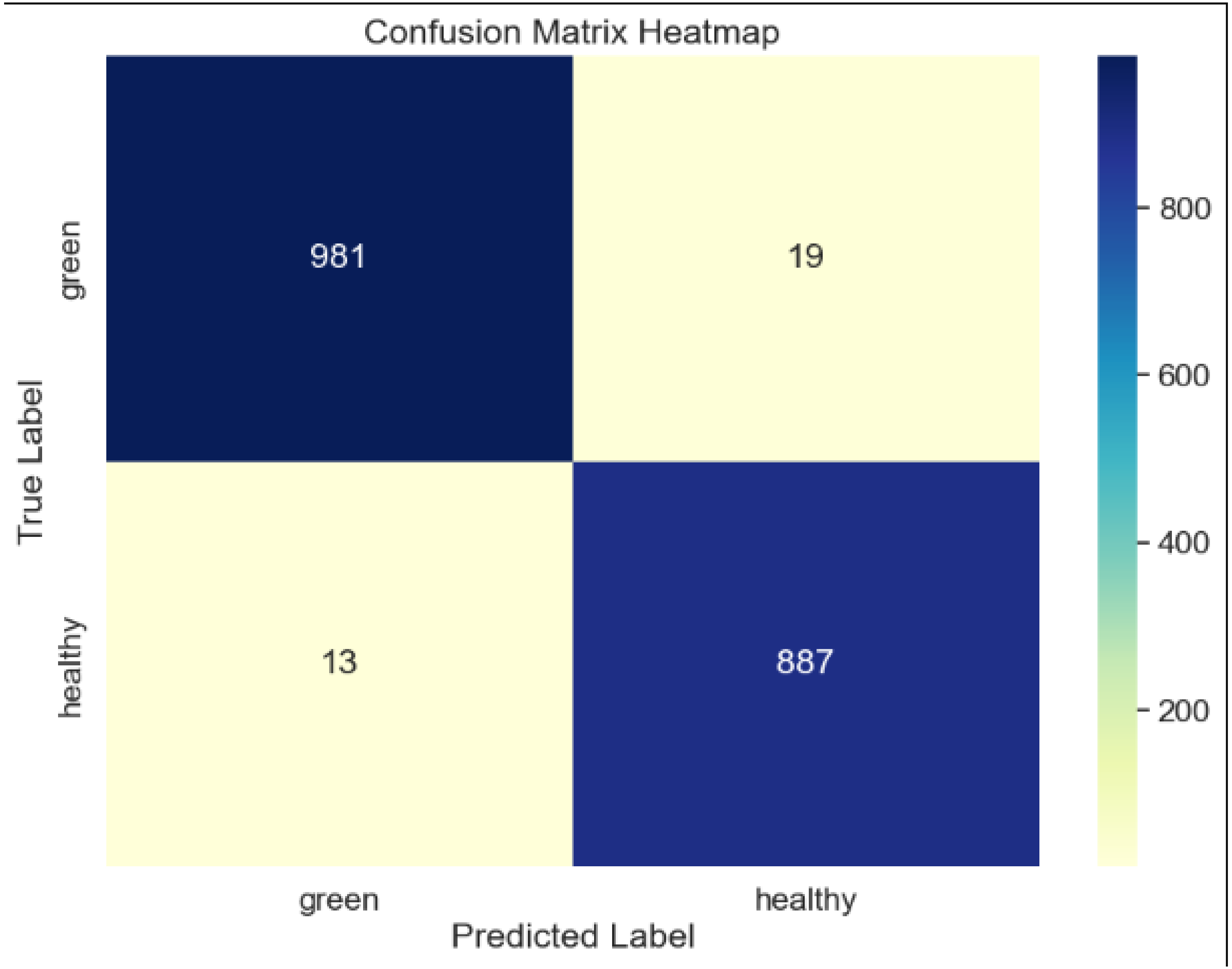
Result: MobileNet [3]
Precision Recall Curve



Result: MobileNet [4]
Precision Recall Curve

- It shows excellent performance for Class 1 (likely HLB-infected class).
- Average Precision (AP) for Class 1 is 0.97, indicating high accuracy in identifying positive cases.
- The curve maintains very high precision (close to 1.0) across most recall values.
- There's a sharp drop in precision only at the very end of the recall spectrum.

Result: MobileNet [5] Confusion Matrix



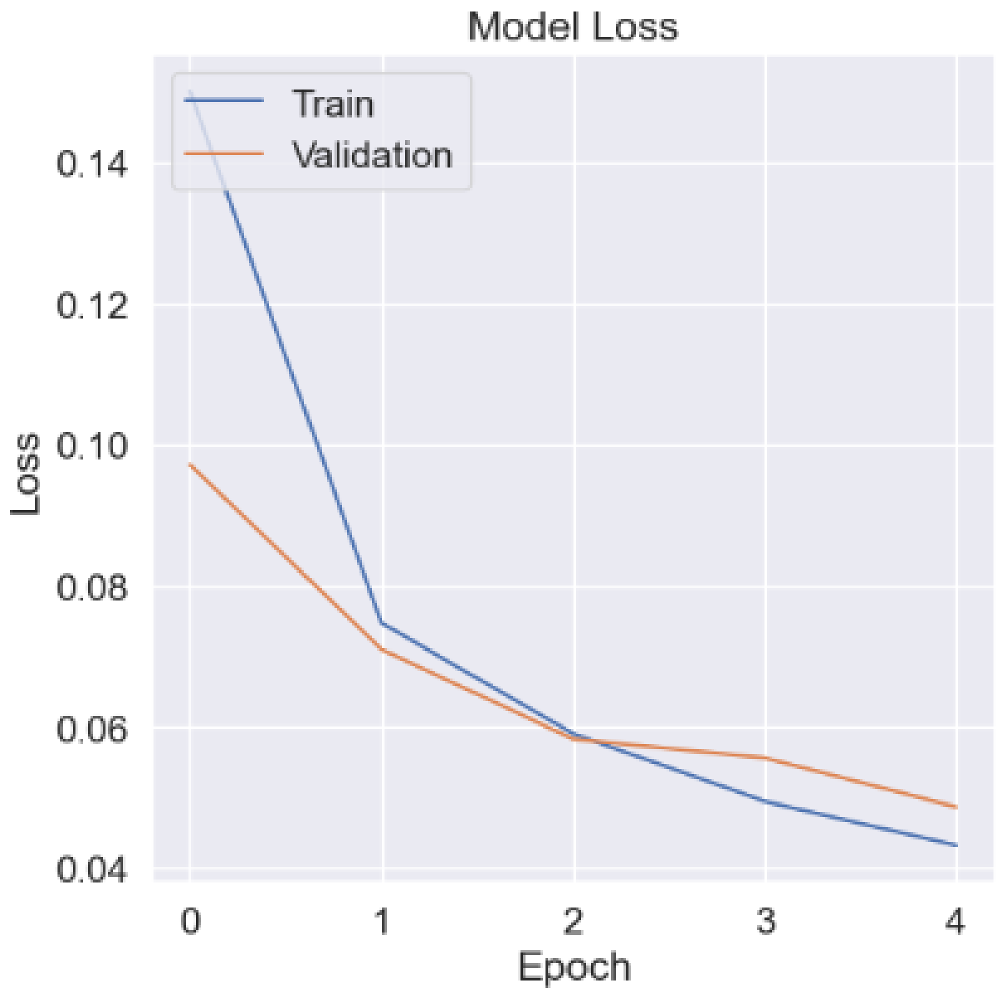
Result: MobileNet [6]
Confusion Matrix

- Shows High Accuracy in classifying both "green" (likely HLB-infected) and "healthy" citrus samples.
- For the "green" class: 981 correct predictions, 19 misclassifications
- For the "healthy" class: 887 correct predictions, 13 misclassifications
- Overall accuracy: (981 + 887) / (981 + 19 + 13 + 887) = 98.31%
- Low false positive and false negative rates for both classes.
- Slight tendency to misclassify healthy plants as infected (19 cases) more often than vice versa (13 cases).

Discussions [1]
ROC,CONF,PR

- A well-trained model that effectively identifies the majority of HLB-infected samples.
- A dataset with clear and distinguishable features indicative of HLB infection.
- A potential slight class imbalance, with fewer positive (infected) samples compared to negative ones.
- Possibly distinct visual markers of HLB that the CNN effectively learned to identify.

Discussion [2]
Accuracy VS Loss Plots of Training and Validation



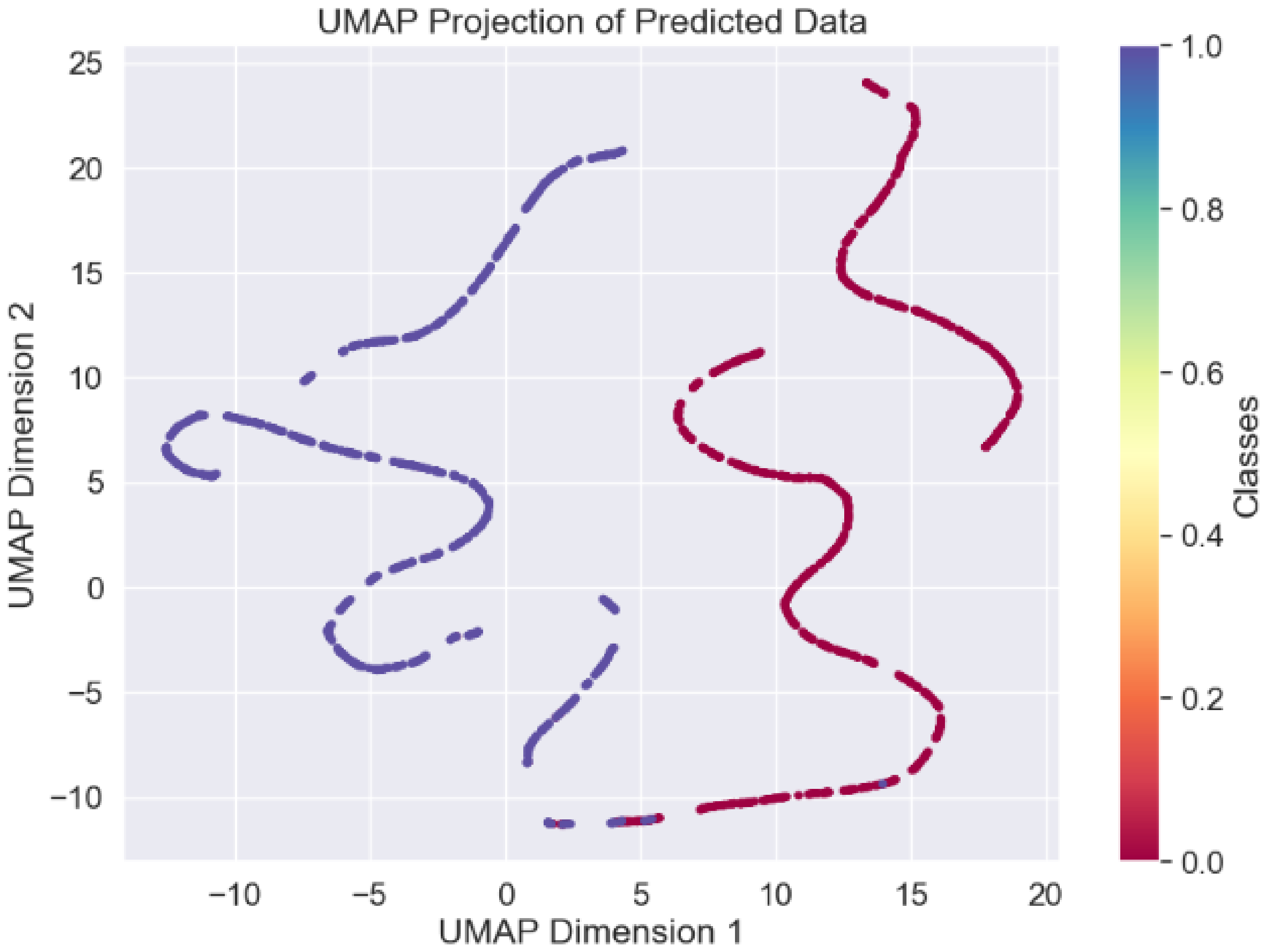
Accuracy VS Loss Plots of Training and Validation[2]

- Accuracy:
  - Training accuracy starts at 0.945 and increases to approximately 0.985 over 4 epochs.
  - Validation accuracy starts at around 0.970 and increases to approximately 0.980 over 4 epochs.
- Loss:
  - Training loss decreases from 0.14 to about 0.05 over 4 epochs.
  - Validation loss decreases from 0.10 to about 0.06 over 4 epochs.

Accuracy VS Loss Plots of Training and Validation [3]

- The model avoids over fitting, showing balanced training and validation performance.
- The reduced loss indicates clean, well-labeled data, improving accuracy.
- The model's consistent rise in accuracy indicates effective feature learning.
- Training for a few more epochs might improve the model further, as the curves have not plateaued completely.

Discussion [3]
UMAP (Uniform Manifold Approximation and Projection)



UMAP (Uniform Manifold Approximation and Projection) [2]

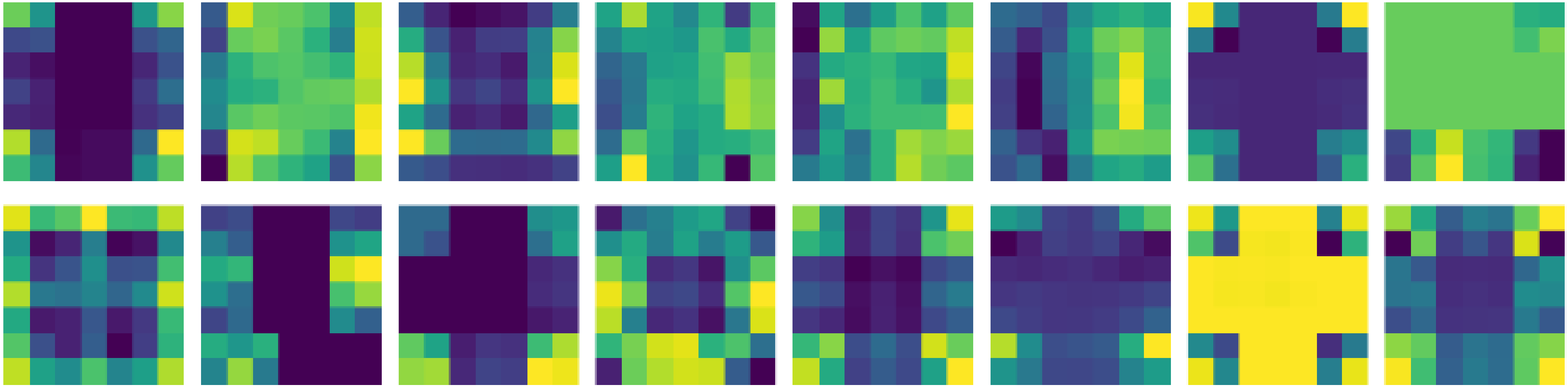
- Color in shades of purple is representing HLB and red shade represent the healthy in the predicted values
- Adequate space between clusters is indicating the model can differentiate between classes
- Datapoints of similar classes lying close to each other is showing the consistency of the model

UMAP (Uniform Manifold Approximation and Projection) [3]

- The distinctive difference between the clusters is showing the images of two classes have different features.
- Well defined separated clusters is suggesting that the model has learned to distinguish between the classes.

Discussion [4]
Feature Map

Feature Maps for the First Image from Layer block_15_depthwise_BN



Feature Map [2]

- Above feature map explains the output of first 10 filters among 96 total filters present in the layer block_15_depthwise_BN.
- Each filters in the layer extracts some specific features from the image.
- Various activation patterns are visible across the feature maps, indicating different regions of importance detected by the network
- The color scale ranges from purple indicating low activation to green/yellow indicating high activation.

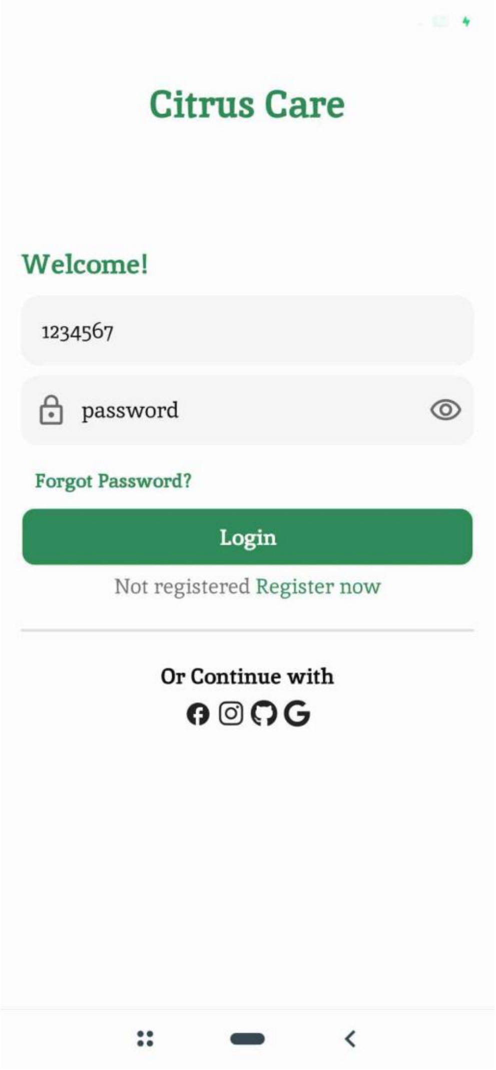
Feature Map [3]

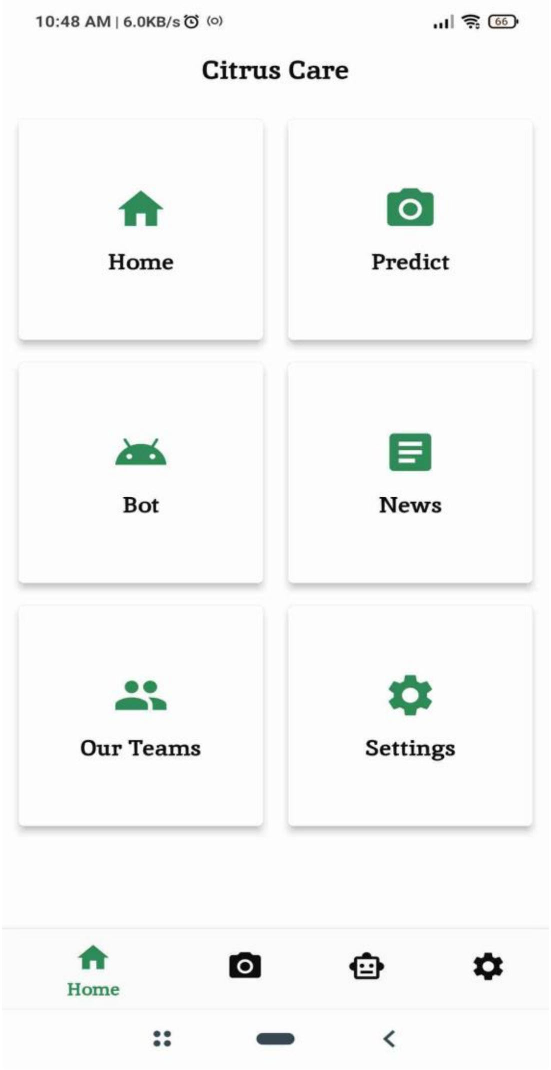
- As feature map is the direct result of training process, network has learned to emphasize certain patterns based on the training data
- block_15_depthwise_BN layer captures the computational patterns and textures in input image.
- To enhance the model further, we can diversify the training data to help the network learn more generalized features.
- Also technique like dropout can be implemented to improve the generalization and reduce overfitting.

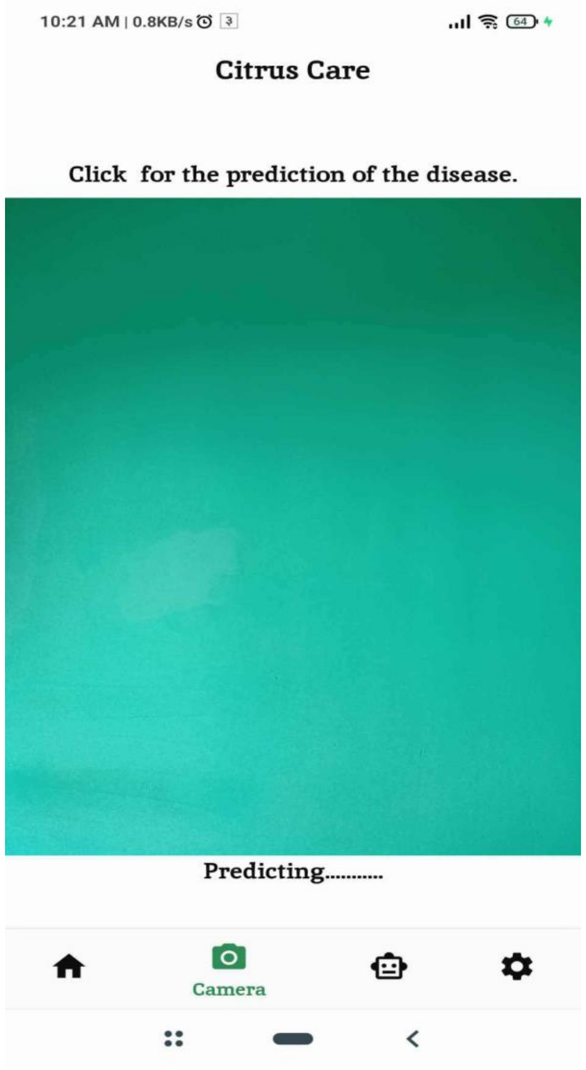
Mobile App

- The cross platform app is built using flutter .
- All the task of the Frontend development have been completed in flutter.
- Camera module for the prediction of the disease has been integrated .
- Backend for the prediction of the disease is in progress.
- Deployment of the application in cloud is on progress.

Mobile App Snapshots







Remaining Workflow

Part A

- Deployment on mobile application locally
- Server
- Generative Adversarial Network (GANs) to compensate imbalanced dataset

Part B

- Ensemble of all the ML models
- ChatBot
- Model Deployment on AWS cloud
- Fully functional app with connection with server
- Field Testing

Thank You!