Early Detection and Prevention of HLB Disease in Citrus Orchards 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

June 20, 2024

Presentation Outline

- Motivation
- HLB Disease
- Objectives
- Project Scope
- Methodology
- Software Requirement
- Expected Results
- Project Applications
- Tentative Planning of Workflow
- Tentative Timeline (Gantt Chart)
- Budget Estimation
- References

Motivation

- HLB is a deadliest bacterial disease which has no cure. So, the early detection the top most priority.
- Due to the poor economic condition of farmers in Nepal, they cannot afford the expensive detection mechanisms.
- So, there is a need of an innovative and effective technique to detect and prevent the spread of HLB.
- This is the reason we took interest in this project

HLB Disease

- HLB (Huanglongbing) or Citrus Greening disease
- Caused by Psyllid nymphs
- Slowly kills citrus tree by obstructing the flow of nutrients

Fig: Symptoms and cause of HLB

Objectives

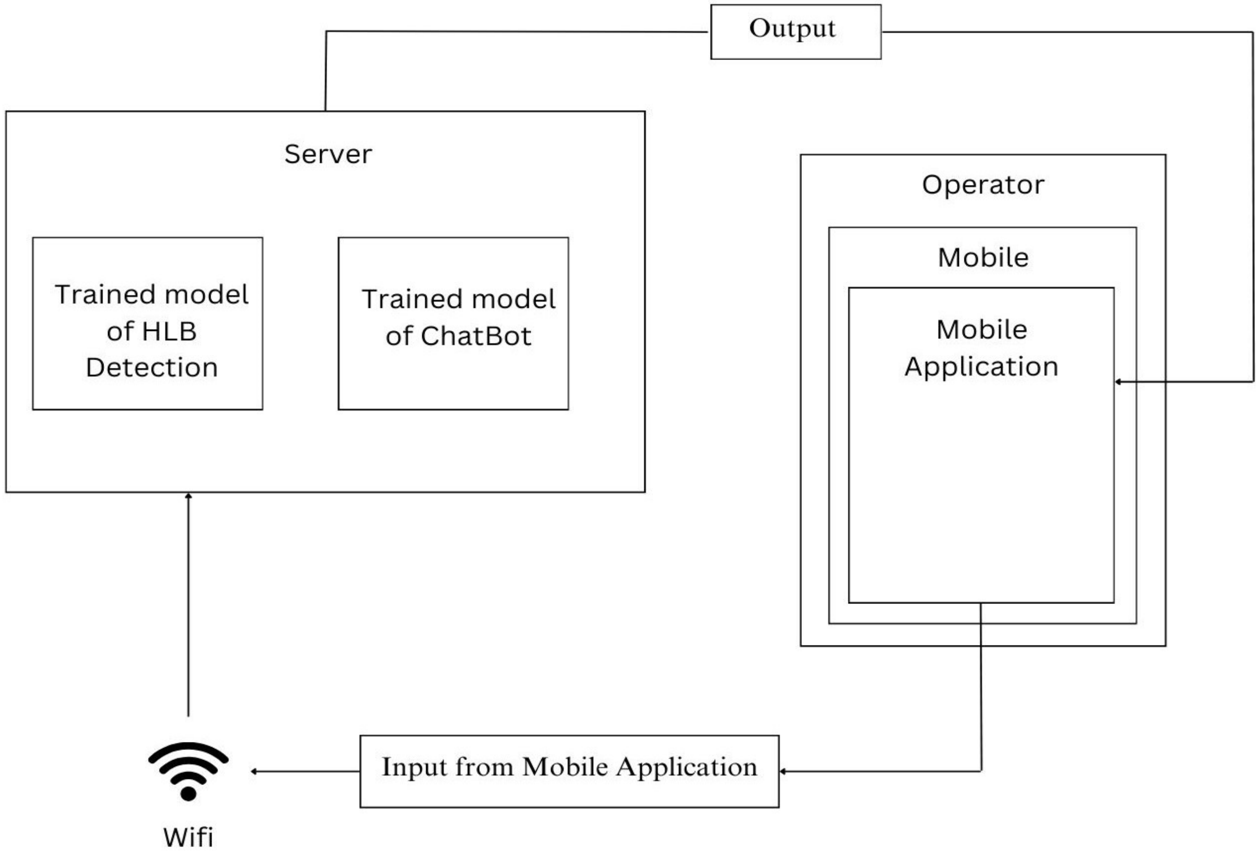
- To train the model for early detection of HLB disease using deep learning
- To create a mobile application that assists in early detection of HLB disease and suggests proper measures for timely prevention.
- To develop a chat bot as a virtual assistant that can help in analyzing, detecting, alerting, and preventing HLB disease along with the information related with the citrus plants.

Project Scope

- Capabilities
  - Detection of HLB disease in citrus plants
  - Chatbot for one to one interaction with the user about HLB detection and the information of citrus plants.

- Limitations
  - The application is only limited to citrus plants.
  - The answering capability of the chatbot is limited.

Methodology [1]
Proposed System Architecture



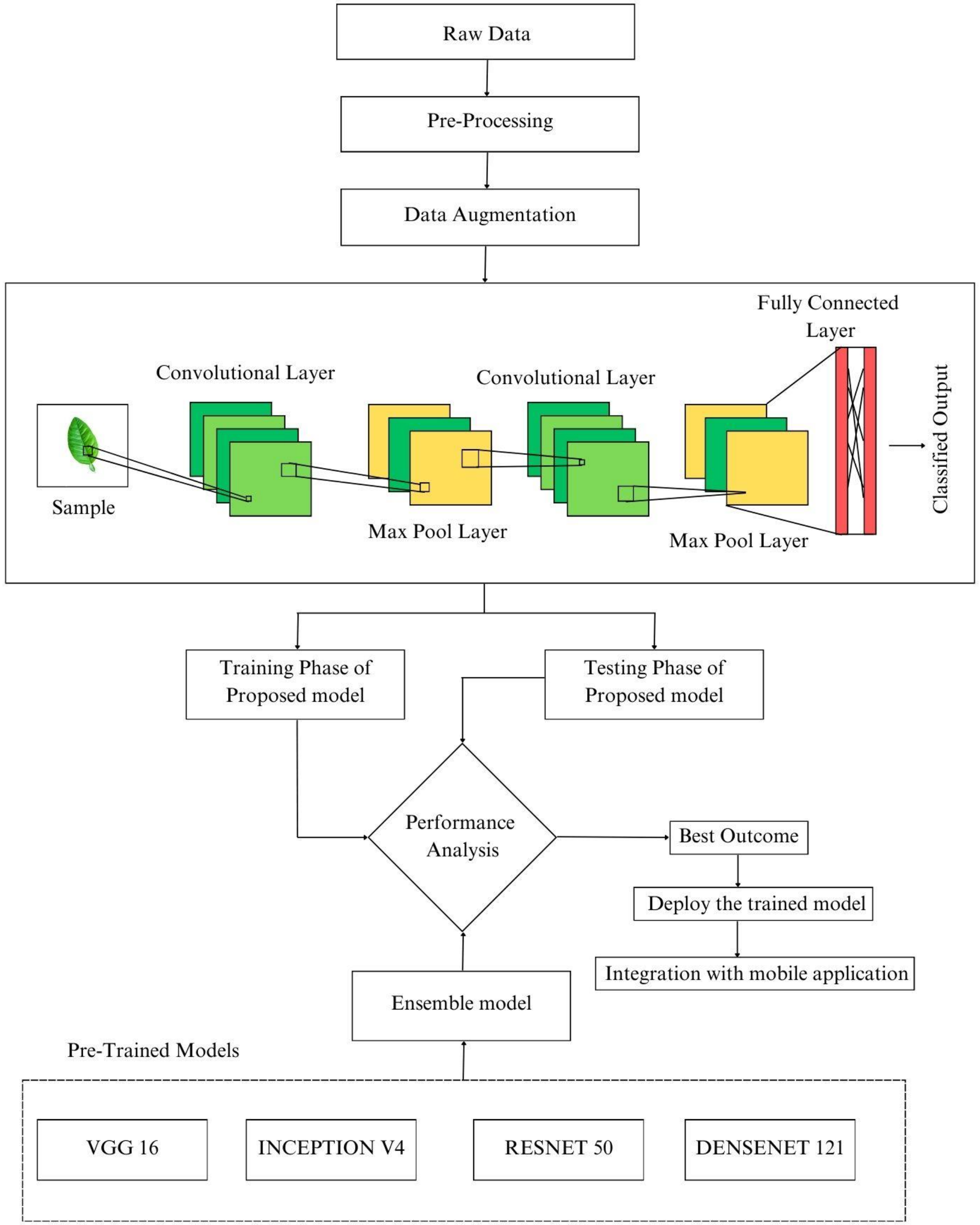
Methodology [2]
App Workflow

- Capture Image of citrus leaves
- Preprocess to ensure it meets the input requirements of the CNN model
- Preprocessed image fed into TensorFlow Lite model to predict the condition of citrus leaf/fruit
- Chatbot provides result and preventive measures if necessary
- Also provides information about Citrus fruits

Methodology [3]
Classification of Dataset

- The dataset is divided into three different classes:
  - Healthy Class
  - HLB Infected Class
  - Other Class
- Images will go under pre-processing steps to improves model's accuracy
- More images for dataset will be collected later
- Dataset for Chatbot will be collected by the team

Methodology [4]
CNN Architecture



Methodology [5]
CNN Training[1]

- Data Preprocessing and Augmentation
- CNN Architecture
- Training
- Loss Calculation
- Model Deployment

CNN Training [2]
Data Preprocessing and Augmentation

- Image resized to 224x224 pixels suitable for CNN
- Scaling of pixels value in range 0-1
- Data augmentation techniques is performed including:
  - Random flips
  - 0-30 degrees rotations
  - Random brightness and contrast adjustments
- Data augmentation increases diversity and model's generalization ability

CNN Training [3]
CNN Architecture [1]

- Consists of 3 convolutional layers with filter counts of 32, 64, and 128 respectively.
- Each convolutional layer will have a 3x3 kernel size.
- The activation function used will be ReLU given as :

$$f(x) = x, \text{ if } x > 0$$

$$f(x) = 0, \text{ if } x \leq 0$$

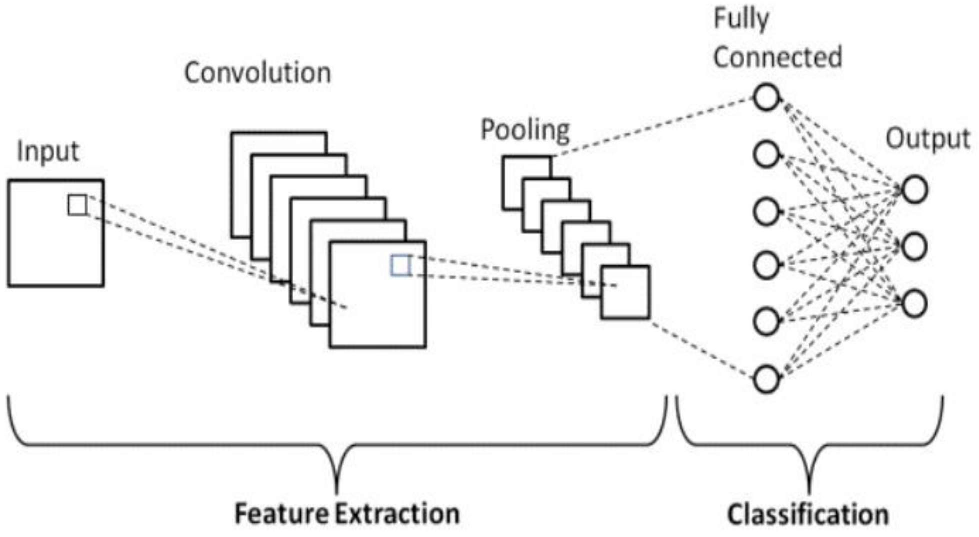


Fig: CNN Architecture

CNN Architecture [2]
Pooling Layer

- Includes 2 max pooling layers with a 2x2 kernel size.
- In each layer, the maximum value is selected as the output
- Purpose:
  - Down sample the spatial dimensions of the feature maps to retain most salient features.
  - Helps to introduce translation invariance, beneficial for image classification tasks

CNN Architecture [3]
Flattening

- **Purpose**: Convert multi-dimensional feature maps into a one-dimensional vector
- **Operation**: Flatten spatial dimensions while preserving learned features
- **Role**: Provide input to fully connected layers for higher-level feature extraction and classification
- **Benefits**: Reduces spatial information, facilitating efficient processing and enhancing pattern extraction

CNN Architecture [4]
Fully Connected Layer

- The CNN architecture will include 2 dense layers, each with 512 units.
- In each dense layer, the ReLU activation function is used for introducing non-linearity
- The purpose of the dense layers is to perform high-level feature extraction
- Also to transform the input data into a suitable format for classification.
- Output layer: 2 units with softmax to classify as HLB or Healthy

CNN Training [4]
Optimizer

- Model uses the Adam optimizer for training.
- It combines the benefits of both the AdaGrad and RMSProp algorithms,
- Provides an adaptive learning rate for each parameter.
- Purpose: achieve efficient and robust convergence during training.
- Other optimizers are not used as it typically offers faster convergence and better performance

CNN Training [5]
Loss Function (Binary Cross Entropy)[1]

- The CNN model will use Binary Cross-Entropy as the loss function.
- Measures the difference between the true labels and the predicted probabilities
- The formula for Binary Cross-Entropy is:

$$L_{BCE} = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot \log \hat{Y}_i + (1 - Y_i) \cdot \log(1 - \hat{Y}_i))$$

Loss Function (Binary Cross Entropy) [2]

- Binary Cross-Entropy quantifies error for two-class problems.
- Other loss functions are not used as:
  - Mean Squared Error is less suitable for binary classification
  - Categorical Cross Entropy is designed for multi-class classification tasks.

CNN Training [6]
Metrices [1]

- The CNN model will be assessed using Accuracy, Precision, Recall, and F1-Score.
- Accuracy gives overall correctness
- Precision assesses positive prediction accuracy.
- Recall measures the model's ability to identify all actual positives.
- F1-Score balances Precision and Recall, providing a comprehensive performance measure.

Metrices [2]
Corresponding Formulas

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{(TP + TN)}{TP + FP + TN + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Where, TP = True Positive; FP = False Positive;

TN = True Negative; FN = False Negative

CNN Training [7]
Regularization

- **Purpose** : To prevent overfitting and improve the model's generalization ability.
- **Dropout (0.5)**: Randomly deactivates 50% of neurons during training
- **Early Stopping**: Halts training when validation performance deteriorates, prevents overfitting

Regularization techniques mitigate overfitting, enhancing model performance and stability

CNN Training [8]
Hyper Parameter Tuning

- **Epochs (50):** Balanced to allow model convergence without risking overfitting.
- **Batch Size (32):** Strikes a balance between introducing noise and computational efficiency.
- **Learning Rate (0.001):** Ensures stable training, avoids divergence or slow convergence

CNN Training [9]
Ensemble Method

- **Combined Model Approach**: Employed to fuse predictions from multiple models, enhancing accuracy.
- **Diverse Model Selection**: Varied models will be chosen to create a diverse ensemble, improving performance.
- **Voting or Averaging**: Predictions will be aggregated using voting or averaging techniques
- **Boosted Performance**: benefit the model to maximize predictive performance.
- **Robustness and Generalization**: It will be improved through ensemble fusion

CNN Training [8]
Deployment

- Convert trained model to TensorFlow Lite format
- Integrate TF Lite model into Flutter mobile app
- Ensure seamless operation for efficient on-device inference

Methodology [6]
Chatbot: NLP [1]

- Natural language processing (NLP)
- A subfield of computer science and artificial intelligence (AI)
- Uses machine learning to enable computers to understand and communicate with human language.
- Can understand even complex human phrasing

Chatbot: NLP [2]

- Text Preprocessing: tokenization, lowercasing, stop word removal, punctuation, stemming
- Feature Extraction: Bag of words, TF-IDF, word embeddings
- Model training: data splitting, model selection, training, evaluation
- Text Analysis: classification, NER, POS tagging, dependency parsing
- Result interpretation

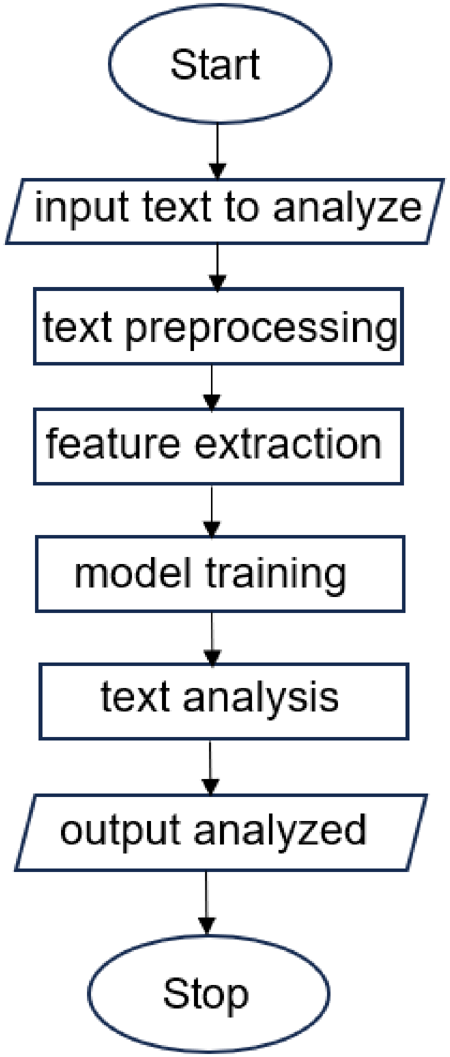


Fig: NLP Flowchart

Chatbot [3]

- Preprocessing: tokenization, lowercasing, stop word removal, punctuation, stemming
- Intent Recognition: feature extraction, classification
- Entity Recognition
- Dialogue Management: context, response, fallback
- Response Generation: template, retrieval, generative
- Post processing

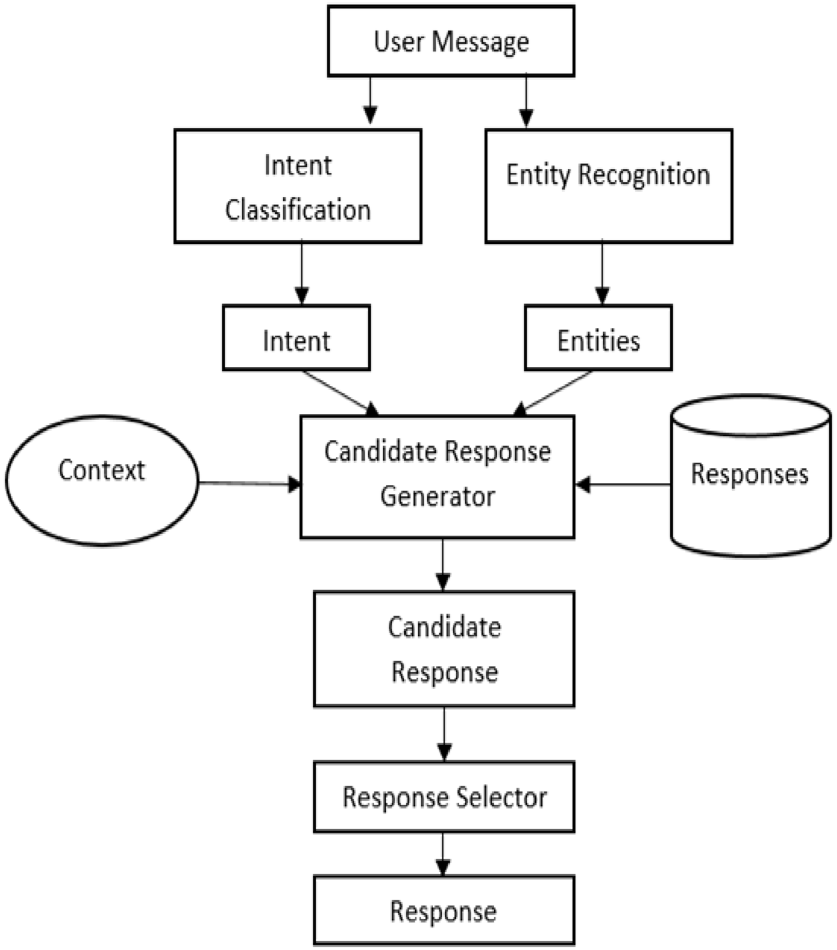


Fig: Chatbot Flowchart

Chatbot [4] Analysis

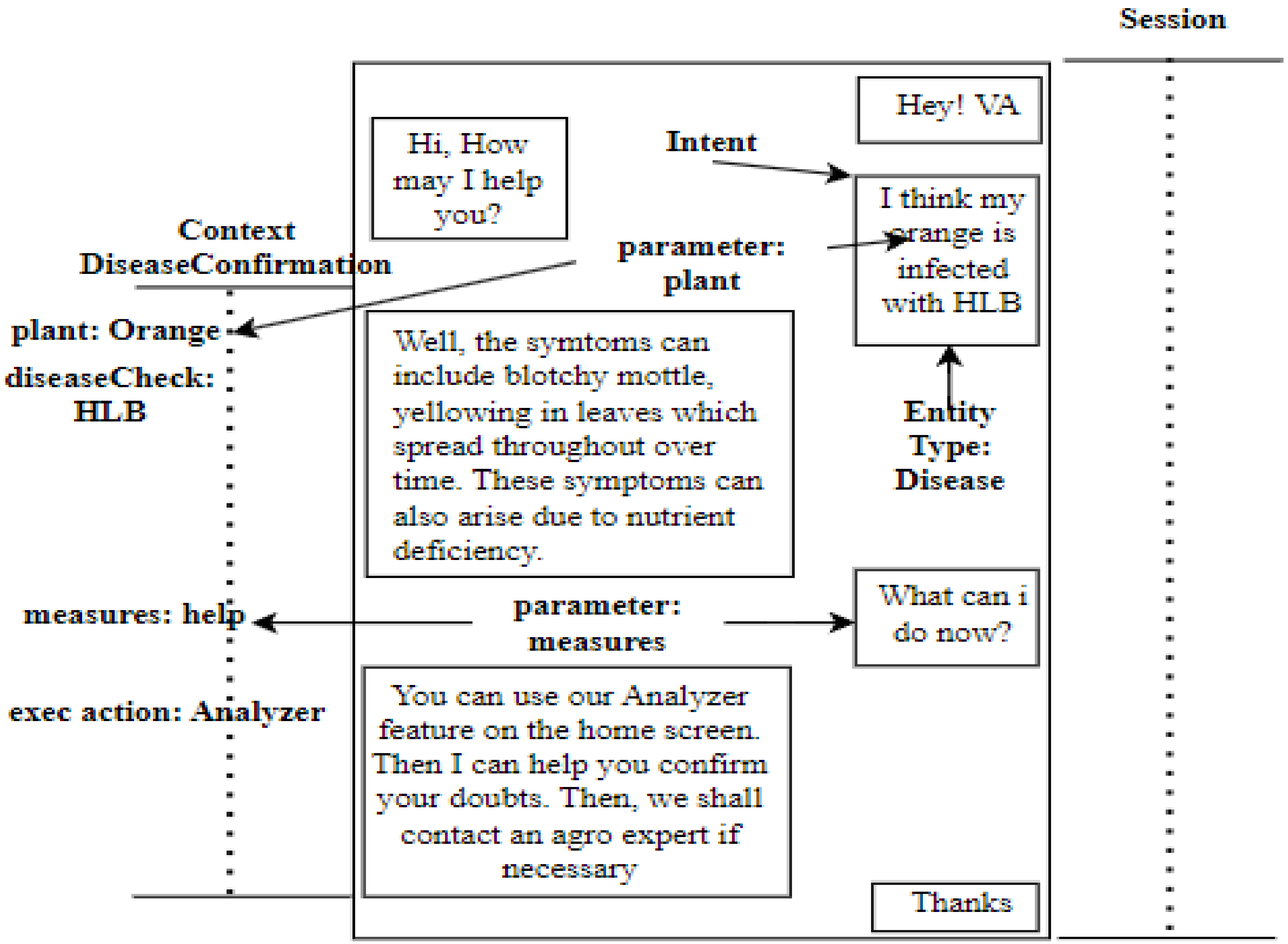


Fig: Chatbot Analysis

Methodology [7]
Mobile Application Development

- The Mobile App is built using Flutter.
- Supports cross - platform with the single code base
- Has Inbuilt chatbot functionality in it

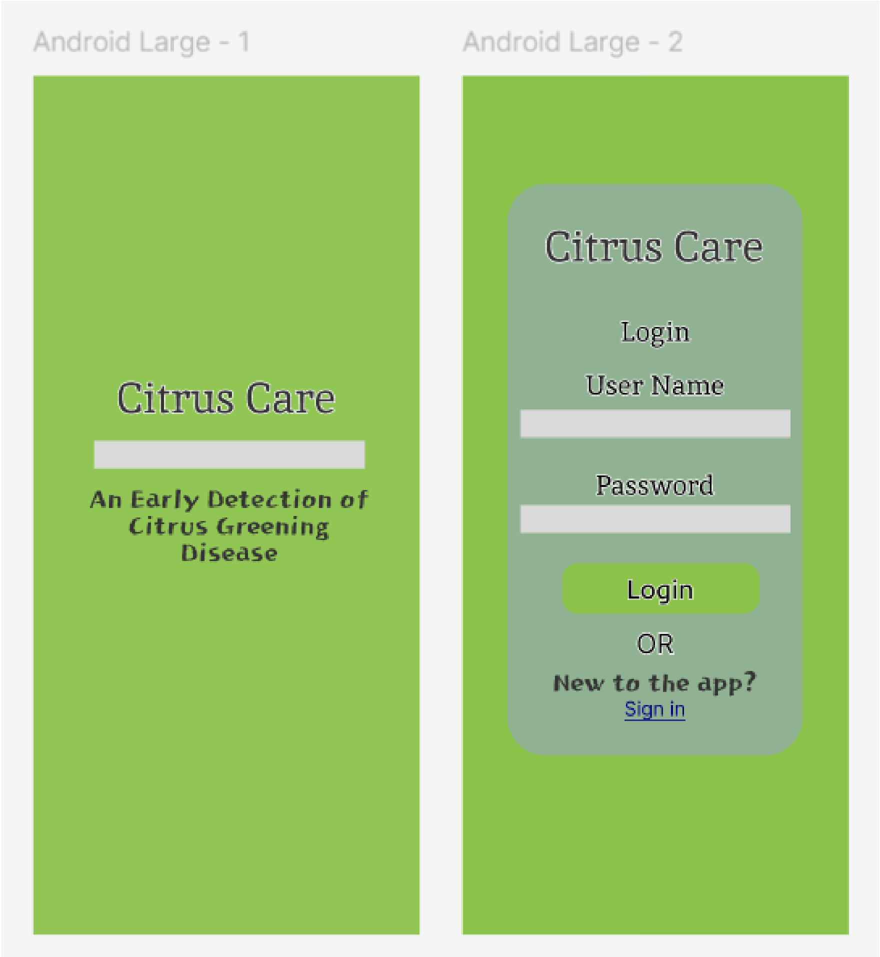
Software Requirement [1]

- Python: For developing machine learning models and backend logic.
- Flutter: For building cross-platform mobile applications.
- Jupyter Notebook: For prototyping machine learning models, and documenting code.
- VS Code: Code editor
- Google Colab: Free cloud-based Jupyter notebooks for model training and experimentation
- AWS Cloud: Hosting machine level models and backend server

Software Requirement [2]

- OpenCV: Library for computer vision tasks
- Pillow (PIL): Python Imaging Library for image manipulation tasks
- Scikit-learn: Tools for data preprocessing, model evaluation, and performance metrics for machine learning tasks.
- Matplotlib and Seaborn: Visualization libraries to analyze model performance and results
- TensorFlow / Keras and PyTorch: For building and training neural networks

Expected Outcomes [1]



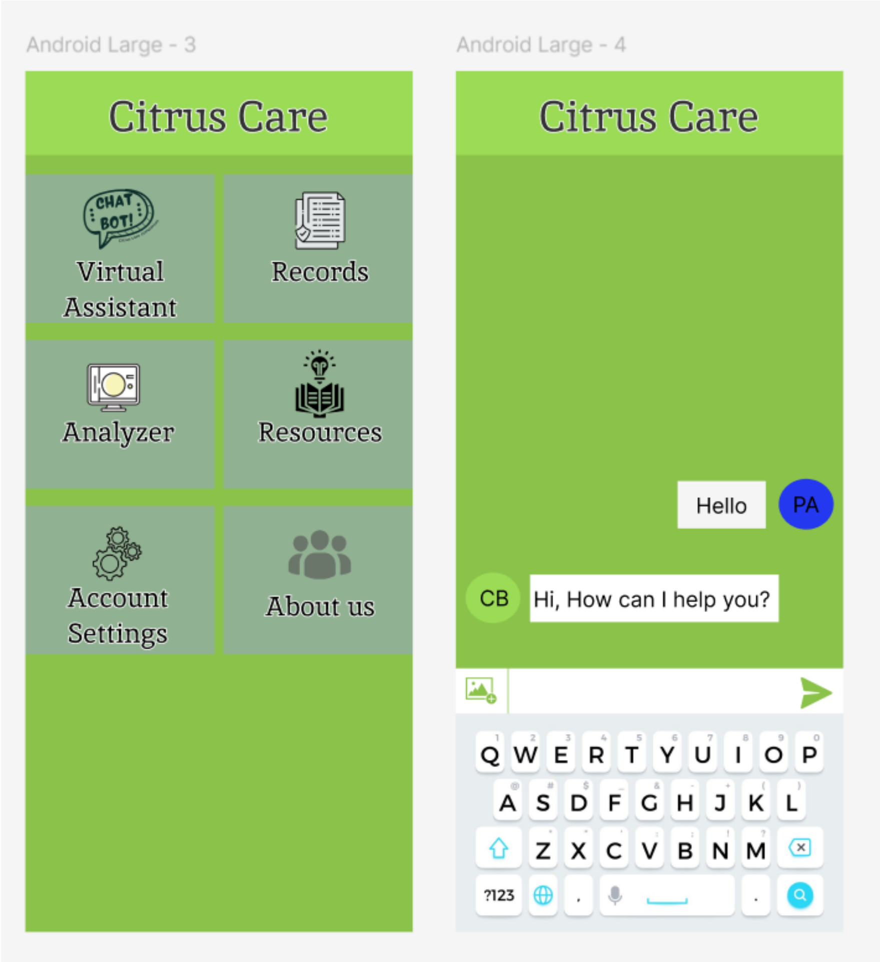
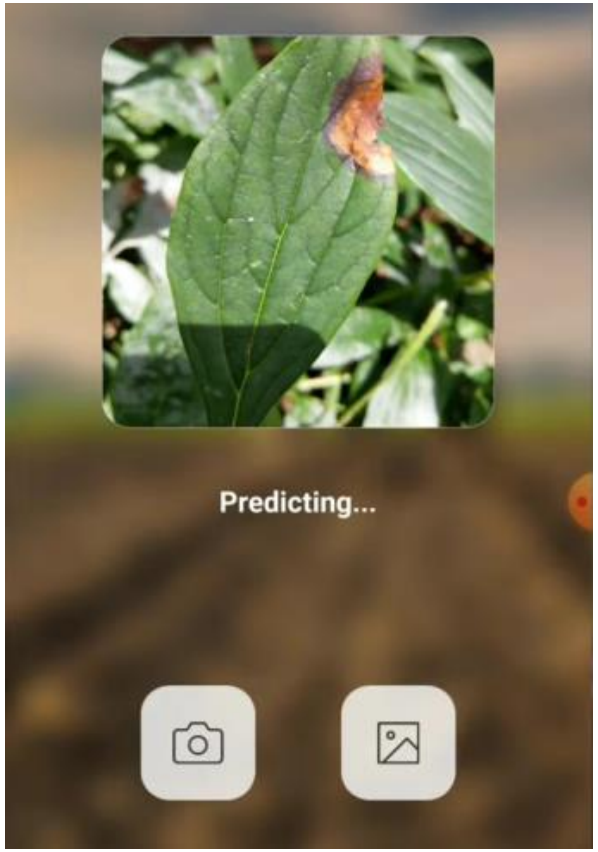


Fig: Expected UI of mobile application

Expected Outcomes [2]



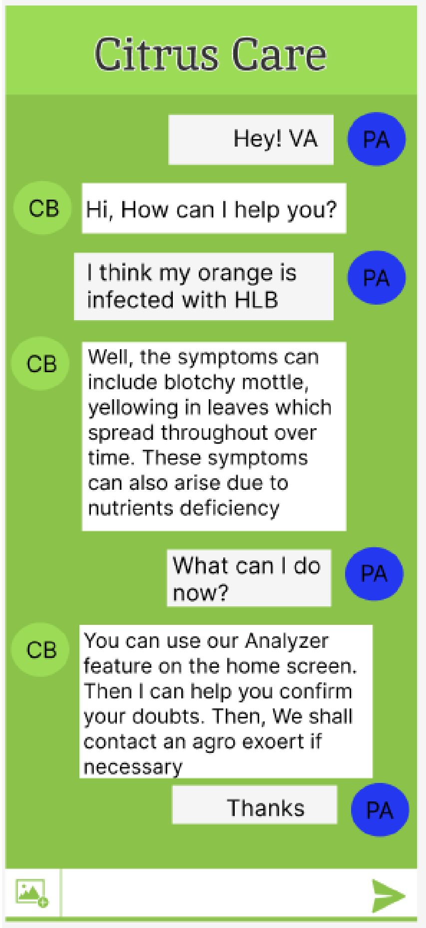


Fig: Expected UI of mobile application

Project Application

- **HLB Detection**: Can be used for early detection of HLB in citrus plants.
- **Monitoring**: Can be used in regular monitoring of the orchards.
- **Altering**: Can also be used in alerting the farmers in case of outbreak in nearby orchards.
- **Prevention**: Can also be used in preventing the HLB disease by providing necessary resources and aids

Tentative Planning of Workflow

Part A

- UI design of app
- Collection of Dataset
- Dedicated data preprocessing
- Individual train of the dataset on 3-4 algorithms
- Server

Part B

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

Tentative Timeline (Gantt Chart)

Estimated Project Expenses/Budget

Items	Price (NRs.)
Hosting Server	Rs. 2000
Data Collection	Rs. 4000
Printing	Rs. 5000
Total	**Rs. 11000**

Reference [1]

- S. Pokharel, S. Pandey, A. Ghimire and S. Kandel, "Understanding Citrus Greening Disease and Its Possible Management," International Journal of Applied Sciences and Biotechnology, vol. 9, pp. 227-238, 2021.
- E. Boa, "Citrus Huanglongbing (Greening) Disease," Plant health cases, April 2023.
- L. Li, S. Shang and B. Wang, "Plant Disease Detection and Classification by Deep Learning—A Review," IEEE Access, vol. 9, pp. 56683-56698, 2021.
- B. Kaur, T. Sharma, B. Goyal and A. Dogra, "A Genetic Algorithm based Feature Optimization method for Citrus HLB Disease Detection using Machine Learning," IEEE Xplore, pp. 1052-1057, 2020.

Reference [2]

- G. Geetharamani and J. A. Pandian, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," Computers & Electrical Engineering, vol. 76, pp. 323-338, June 2019.
- M. Chohan, A. Khan, R. Chohan, S. H. Katpar and M. S. Mahar, "Plant Disease Detection using Deep Learning," International Journal of Recent Technology and Engineering, vol. 9, no. 1, pp. 909-914, May 2020.
- D. Yang, F. Wang, Y. Hu, Y. Lan and X. Deng, "Citrus Huanglongbing Detection Based on Multi-Modal Feature Fusion Learning," Front. Plant Sci., vol. 12, 23 December 2021.

Thank You!