

Mobile Computing Project Presentation

Himanshu Kumar

2022215



INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI



- App Name : plant Diseases Detection with Solution
- Key Feature :
 - Diseases Detection
 - LLM-advisory System
 - Offline based articature



Application Overview



- Core component
 - `MainActivity`: Entry point with theme management
 - `ModelFileHelper`: Model initialization handler
 - `DetectionEngine`: PyTorch Mobile integration
 - `ChatSystem`: LLM-powered advisory



Key Activities



- MainActivity
 - Handle app initialization
 - Theme State
 - Navigation Controller
- DiseaseDetectionScreen
 - Camera/Gallery integration
 - Image Preprocessing Pipeline



- Disease Detection Flow
 - Image capture
 - Bitmap preprocessing
 - PyTorch model inference
 - Softmax probability calculation
 - Multi-threshold result display



- PlantDiseasesModel.pt:
 - ResNet-50 base architecture
 - 38-class output layer
 - Optimized for mobile
- Class Mapping:
 - JSON-based label system
 - 45+plant species support



Ai Processing Pipeline



- suspend fun detectDisease() {
 - 1. loadModel() // PyTorch Mobile
 - 2. preprocessImage() // Tensor conversion
 - 3. model.forward() // GPU-accelerated
 - 4. softmax() // Confidence calculation
 - 5. formatResults() // Top-3 predictions
- }



Advanced Features



- Nlp based solution:
 - Offline Mode : static Recommendations
 - openAI : GPT-3.5-turbo // we can't use for now due to free api limit exceed
 - Local LLMs: Llama-2/phi-2
- Dynamic Thresholding:
 - User-configurable confidence (10-90%)



Chat System Architecture



- ChatScreen Components:
 - MessageREcycleView
 - OpenAi API client
 - Message persistence
 - Context-aware routing:
 - `when {`
 - `containsDiseaseKeywords() -> LocalLLM()`
 - `else -> OpenAI()`
 - `}`



Technical Stack

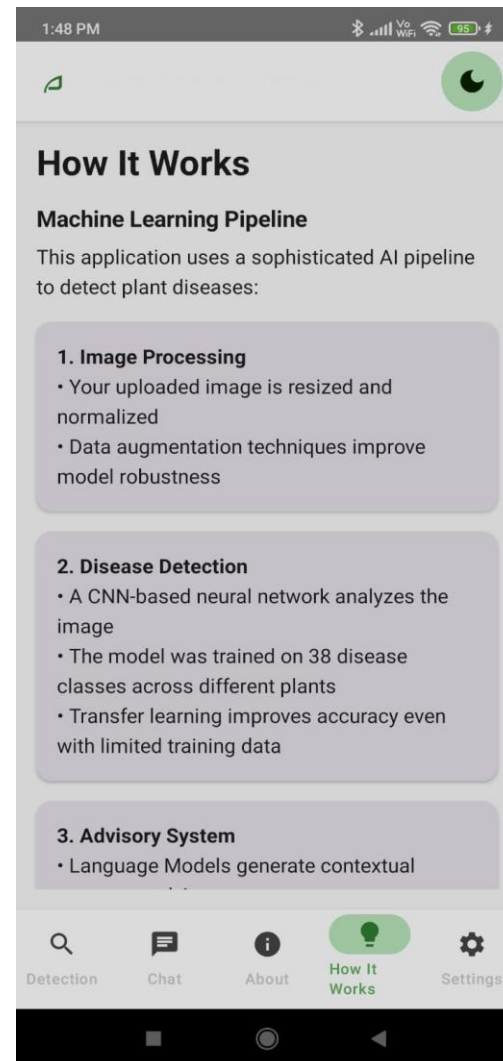
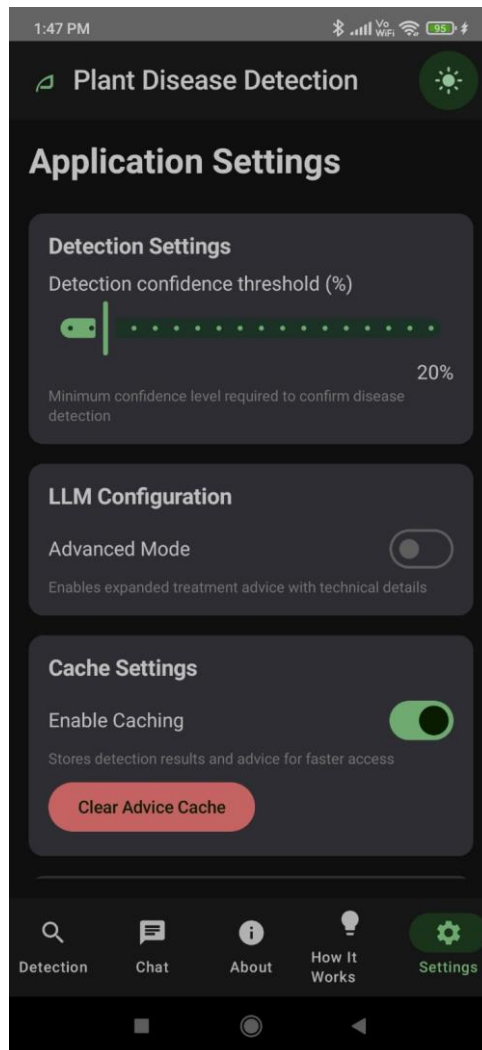


- Frontend:
 - Jetpack Compose Ui
 - CameraX API
 - Navigation Component
- Backend:
 - PyTorch Mobile
 - OpenAI Kotlin SDK

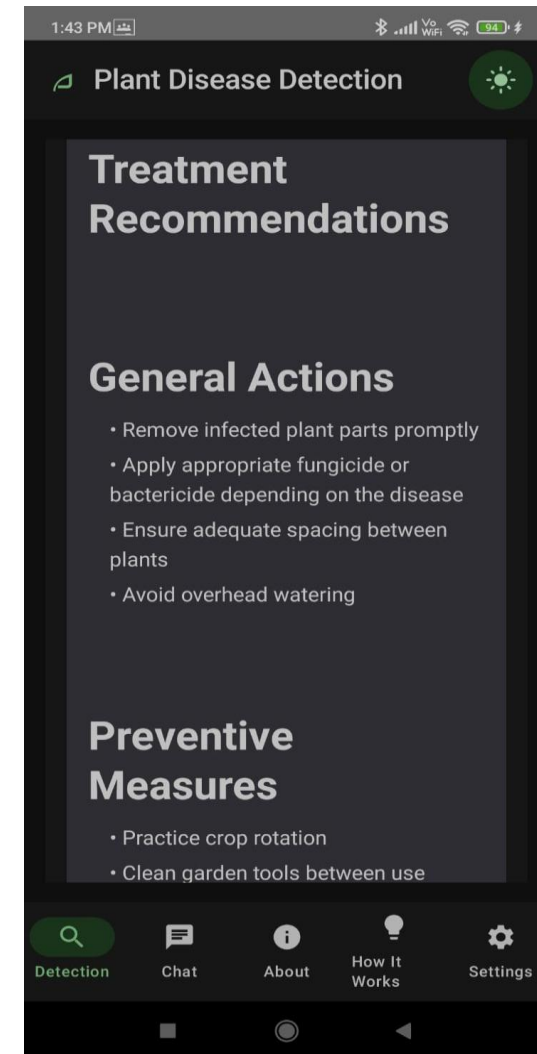
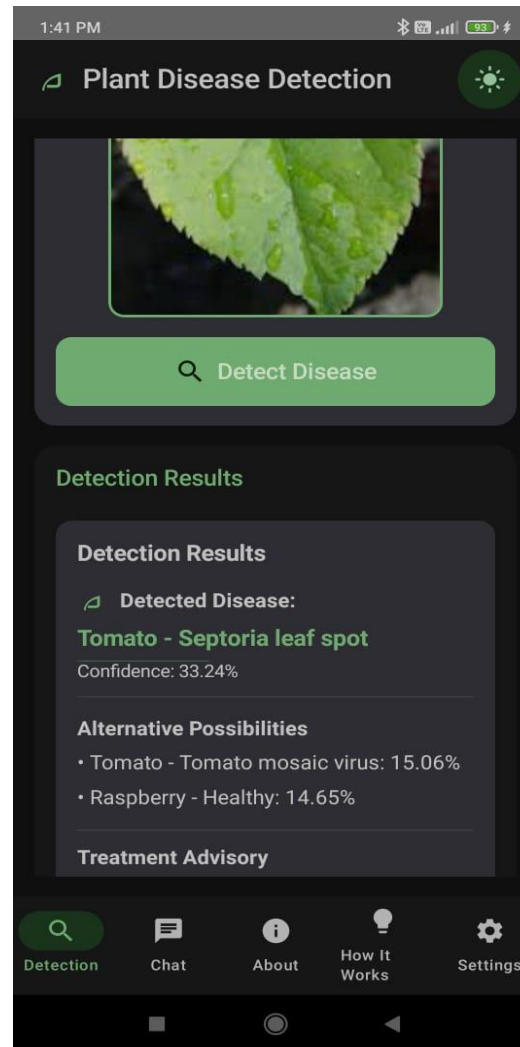
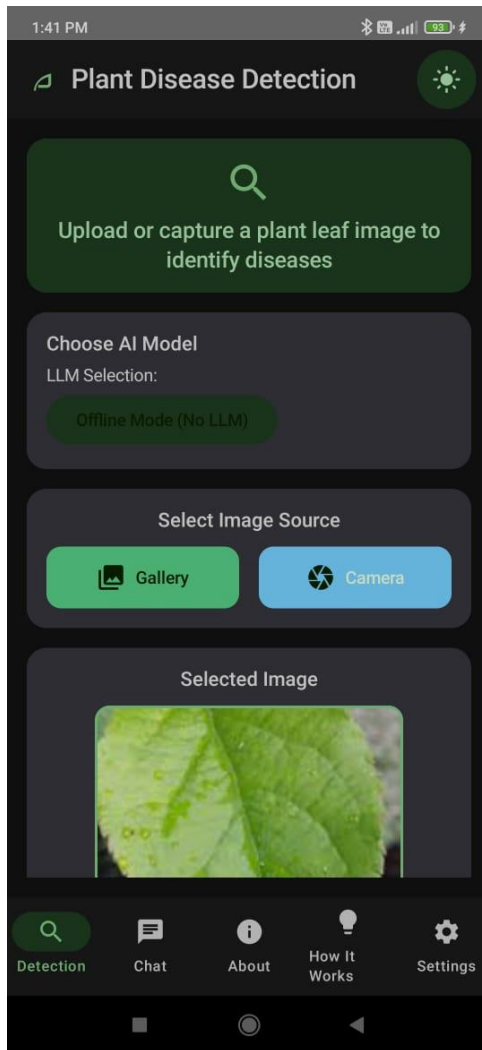


- Model Metrics:
 - 96% Test Accuracy
 - We run test on dataset which is similar to train dataset . So there may be difference in accuracy in different dataset.
- App Size:
 - 450 mb
 - Mostly due to offline model .
 - We can use online model . But here we got latency

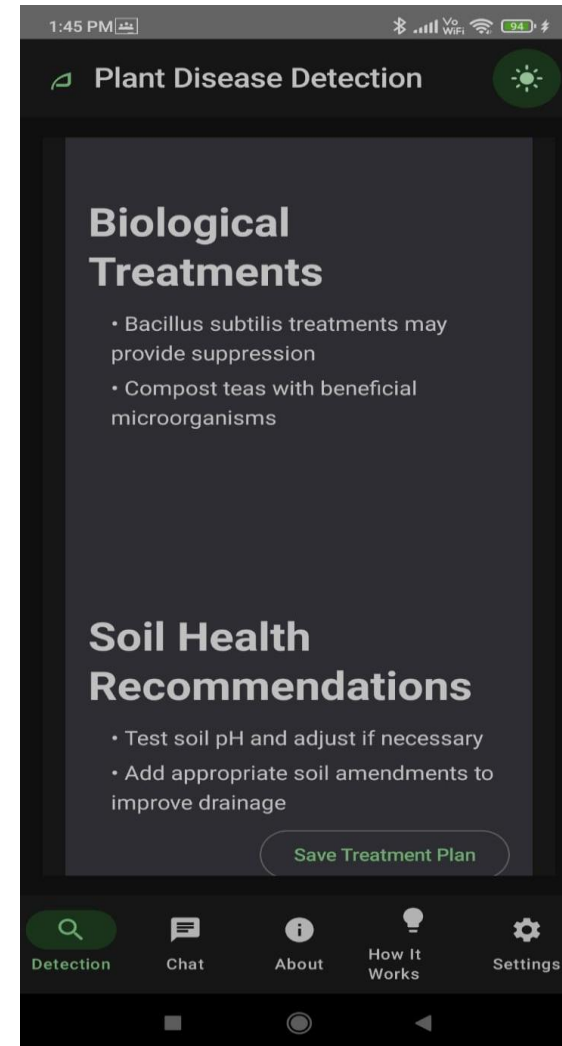
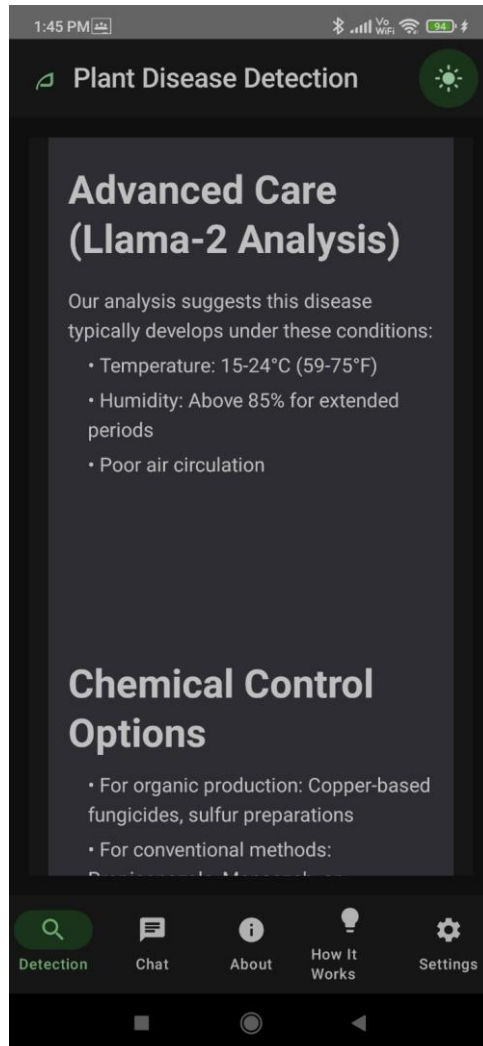




Testing Result



Testing Result



Challenge Solved



- Diseases detection in android app
- Camera/Gallery permission Handling
- LLL Response Caching
- Multi- Theme Support



Future Roadmap



- Short-Term :
 - Real-Time Camera Analysis
 - Diseases Progression Tracking
- Long-term:
 - AR Visualisation
 - Satellite Integration
 - Global Diseases Map
 - Farmer Community Features

