AI BASED CROP IDENTIFICATION MOBILE APP

A MINI PROJECT REPORT

submitted

in the partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

B. SAI HEMANTH REDDY (17B81A05H5)

T.P.N. NIKHIL (17B81A05F2)

D.V. KRISHNA KALYAN (17B81A05C2)

Under the guidance of

MS. M. SATHYA DEVI
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

(An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Rangareddy (D), Telangana- 501 510

June 2020

AI BASED CROP IDENTIFICATION MOBILE APP

A MINI PROJECT REPORT

submitted

in the partial fulfillment of the requirements for the award of the degree

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

by

B. SAI HEMANTH REDDY (17B81A05H5)

T.P.N. NIKHIL (17B81A05F2)

D.V. KRISHNA KALYAN (17B81A05C2)

Under the guidance of

MS. M. SATHYA DEVI
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CVR COLLEGE OF ENGINEERING

(An Autonomous institution, NBA, NAAC Accredited and Affiliated to JNTUH, Hyderabad)

Vastunagar, Mangalpalli (V), Ibrahimpatnam (M),

Rangareddy (D), Telangana- 501 510

June 2020

Cherabuddi Education Society's

CVR COLLEGE OF ENGINEERING

(An Autonomous Institution)

ACCREDITED BY NBA, NAAC 'A' GRADE

(Approved by AICTE & Government of Telangana and Affiliated to JNTU Hyderabad) Vastunagar, Mangalpalli (V), Ibrahimpatnam (M), R.R.District.

Web: http://www.cvr.ac.in, email: info@cvr.ac.in

Ph : 08414 – 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"AI BASED CROP IDENTIFICATION MOBILE APP"** that is being submitted by **B. SAI HEMANTH REDDY (17B81A05H5)**, **T.P.N. NIKHIL (17B81A05F2)**, **D.V. KRISHNA KALYAN (17B81A05C2)**, in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering to the CVR College of Engineering, is a record of bonafide work carried out by them under my guidance and supervision during the year 2019 - 2020.

The results embodied in this project work has not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of the project guide

MS. M. SATHYA DEVI

Associate Professor

CSE Department

CVR College of Engineering

Signature of the HOD

DR. K. VENKATESWARA RAO

Head of the Department

CSE Department

CVR College of Engineering

External Examiner

Cherabuddi Education Society's
CVR COLLEGE OF ENGINEERING
(An Autonomous Institution)
ACCREDITED BY NBA, NAAC 'A' GRADE
(Approved by AICTE & Government of Telangana and Affiliated to JNTU Hyderabad) Vastunagar,
Mangalpalli (V), Ibrahimpatnam (M), R.R.District.
Web: http://www.cvr.ac.in, email: info@cvr.ac.in
Ph : 08414 – 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that the project entitled **"AI BASED CROP IDENTIFICATION MOBILE APP"** that is being submitted by **B. SAI HEMANTH REDDY (17B81A05H5)**, **T.P.N. NIKHIL (17B81A05F2)**, **D.V. KRISHNA KALYAN (17B81A05C2)**, in partial fulfillment for the award of Bachelor of Technology in Computer Science and Engineering to the CVR College of Engineering, is a record of bonafide work carried out by them under my guidance and supervision during the year 2019 - 2020.

The results embodied in this project work has not been submitted to any other University or Institute for the award of any degree or diploma.

Signature of the project guide	Signature of the HOD
MS. M. SATHYA DEVI	**DR. K. VENKATESWARA RAO**
Associate Professor	Head of the Department
CSE Department	CSE Department
CVR College of Engineering	CVR College of Engineering

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible and whose encouragement and guidance has been a source of inspiration throughout the course of the project.

It is a great pleasure to convey our profound sense of gratitude to our Principal **Dr. Nayanathara K. S** and **Dr. K. Venkateswara Rao, Head of the CSE Department**, CVR College of Engineering for being kind enough for arranging the necessary facilities for executing the project in the college.

We would like to express our sincere gratitude to our guide, **Ms. M. Sathya Devi, Associate Prof., CSE Dept.,** CVR College of Engineering, whose guidance and valuable suggestions have been indispensable to bring about the successful completion of our project.

We wish to acknowledge special thanks to the Project Coordinator **Mrs. P. Madhavi, CSE Dept.,** for assessing seminars, inspiration, moral support and giving us valuable suggestions in our project.

We would also like to express our gratitude to all the staff members and lab faculty, department of **Computer Science and Engineering, CVR College of Engineering** for the constant help and support.

We wish a deep sense of gratitude and heartfelt thanks to management for providing excellent lab facilities and tools. Finally, we thank all those whose guidance helped us in this regard.

ABSTRACT

India is an agricultural economy, with most of the population's livelihood is directly or indirectly associated with farming. Since late 1960s technology has advanced at an exponential rate but that advancements haven't improved farming. With the pressure to move to sustainable, green future, we need to advance farming techniques to improve energy requirements for farming, its carbon footprint, etc.

This project is an attempt at solving a small piece of that puzzle. Using AI to provide farmers with information, suggestions, precautions, and educate farmers, we can empower the farmers to help themselves.

The proposed model is currently targeted to support farmers, to educate them, give them information about their crops and diseases. This is created such that it can improve over time with new additional data and is developed using real-world data

TABLE OF CONTENTS

LIST OF FIGURES

I INTRODUCTION

1.1 MOTIVATION

Intelligence has been considered as the major challenge in promoting economic potential and production efficiency of precision agriculture. Modern agriculture seeks to manage crops in controlled environments that are able to improve the production of plants or duplicate the environmental conditions of specific geographical areas to obtain imported products locally.

Now it is possible to obtain highly accurate status of crops and form reasonable decisions to manage irrigation, enrich the soil nutrition in agricultural scenes. Automatic Plant Image Identification is the most promising solution towards bridging the botanical taxonomic gap, which receives considerable attention.

As the technology advances, sophisticated models have been proposed for automatic plant identification. With the popularity of technology, by the use of smartphones, many plant photos have been acquired. This data can be analysed and used to identify a plant. Improving the performance of the mobile-based plant identification helps users identify the crop and obtain the information and geolocation of the respective crop.

Government agencies and agricultural managers require information on the spatial distribution and area of cultivated crops for planning purposes. Agencies can more adequately plan the import and export of crop there by reducing price changes and stress on consumers.

1.2 PROBLEM STATEMENT

Develop a mobile application that can identify crop using the field photo of the crop. The application allows the user to take photos and automatically detects the crop. The photo of the crop along with its information and geolocation, are stored in a database. To ensure farmers regarding latest methodologies and techniques in the process of cultivation and suggest the necessary remedies for the diseases in crops.

1.3 PROJECT REPORT ORGANIZATION

In this *"Project Report"s* a detailed description about the design challenges, proposed methodologies and the implementation of application to solve the real world problem is given. Different functionalities of the application are broken down into modules and explained with the help of use case diagrams and class diagrams. The working model of the application is shown using screenshots in this report.

This report is organized into six chapters. After this introductory chapter,

Chapter 2: Describes the characteristics of the problem, design challenge and proposed solution

Chapter 3: Provides software requirements which includes functional requirements, non-functional requirements, high level architecture of the proposed system and system specifications which includes software requirements and hardware requirements.

Chapter 4: Can be elaborated either in top down number or bottom up manner based on the development strategy adapted. It describes usecase diagrams, class diagrams, activity diagrams, architecture diagrams.

Chapter 5: In this chapter, implementation and testing is discussed in detail.

Chapter 6: Describes the conclusion and future enhancements.

II PROPOSED MODEL

2.1 INTRODUCTION TO THE CHARACTERISTICS OF THE PROBLEM

Though still in the beginning of its journey, ML-driven farms are already evolving into artificial intelligence systems. At present, machine learning solutions tackle individual problems, but with further integration of automated data recording, data analysis, machine learning, and decision-making into an interconnected system, farming practices would change into with the so-called knowledge-based agriculture that would be able to increase production levels and products quality.

The main characteristics include:-

Crop Detection: This feature allows the automated machines in future to differentiate the crops and suggest appropriate techniques in improving crop quality.

Disease Detection: The current existing method for plant disease detection is simply naked eye observation which sometimes may not give accurate results.With the help of ML the past disease remedies can be applied and get immediate solutions to prevent further catastrophe in farming.

2.2 DESIGN CHALLENGES

Creating an Accurate AI Model: The primary design challenge is to creating an AI model with acceptable accuracy. CNN can be used for image detection. A simple NN can be created using transfer learning. This base model is essential to design an Minimum Viable Product (MVP). This MVP is essential for data collection for improving the AI model with Data Engine.

Managing Design Influence: Software is know for its high plasticity and the software requirements changes as stakeholders for the app change. This need to be primary consideration for design of any API/Data Models that are crucial for app's performance.

Ethical Practices: Deep learning model are inherently data hungry and these also tend to perform better with more data. So data collection is essential but this also presents an ethical question about privacy. So, any data collection should only be to improve the AI models and not breach user privacy. Steps need to be take to educate user's what kind of data is collected.

Performance: Any app need to be fast and scalable. All computationally intensive processes that need to be done of user's device need to be run in the background to prevent app from becoming unresponsive. This can be archived using threads. A thread of computation can be created for any heavy computational or network activity. This asynchronous nature can cause race conditions. This race conditions can be dealt with thread safe data models and semaphores.

Scalability: Backend also need to scale based on usage, traffic. So care need to be taken to design API/Date models that can easily scale in realtime based on requirements. Containerizing the backend makes it easy for scalability. Docker is an software environment that can be used to create, maintain and run the containers.

2.3 PROPOSED SOLUTION

Considering the limitations of existing systems, it seems obvious that we need to have a better system. We have developed a simple mobile application and helps farmers with learning about their crops, diagnose any crop diseases without having to waste their resources on futile solutions.

Automatic Plant/Disease Identification is the most promising solution that can be used to partially solve this problem. Solving the entire problem requires an update of basic infrastructure and education. Saying that, this is a small starting step to that future.

Improving the performance of the Mobile-based plant identification helps users identify the crop and obtain the information about the crop/crop disease.

III REQUIREMENTS AND SPECIFICATIONS

3.1 SOFTWARE REQUIREMENTS

3.1.1 Functional Requirements

1. User should be able to login using Email/Google account.
2. User should be able to take/use a picture to get predictions about the crop/crop disease.
3. Save the predictions in an database and access it from other devices.
4. Bookmark predictions.
5. Generate Report using the predictions.
6. Save report as PDF and share it easily using email or chat application (like iMessages, WhatsApp).

3.1.2 Non-Functional Requirements

Performance: Performance is essential for user experience. Performance is critical in both mobile app and backend. Slow performance is detrimental is app usage. With the help of optimization techniques, the entire processing pipeline is tested using strictly timed unit tests.

Scalabilty: This application is built to be have thousands of concurrent users. So, this project was developed with scalability in mind. The server is built on flask which is light-weight, requiring less resources. The web server is containerized, so, depending on the usage, docker can automatically start or stop instances.

Usability: In mobile app, localization support allow users to use the app in their preferred local language. Scalable, optimized backend means low latency for information processing, less downtime and better usability.

Availability: This app is available for both iOS and Android. Since android platform is omnipresent in mobile market, user penetration can easily be archived. The small share of iOS market is also not left alone. In, later versions, image classification can be performed on-device, making internet not a necessity.

Security: All requests to server are secured using TLS. User data is saved on device using on-device encryption. Data on firebase is automatically using user password.

Cost and Maintainability: Firebase is a pay-as-you-go model, lowering the costs considering that maintenance costs are essentially non-existential.

3.2 SYSTEM SPECIFICATIONS

3.2.1 Software Specifications

Pytorch: PyTorch is an open-source machine learning library based on Torch library. PyTorch is a Tensor computing library and also a deep learning library with automatic differentiation system. In can be used to build state-of-the-art Neural Networks which are used in self-driving cars.

The crop detection and crop disease detection is done using a Convolutional Neural Network(CNN) built using ResNet50 as a feature extractor. The current models have an accuracy of 92% and 90%.

NumPy: NumPy is a numerical library that is used for accelerated tensor operations. This help is speeding up the computation by taking advantage of SIMD instructions on modern CPUs and GPUs. This library is useful for preprocessing data before classification.

Matplotlib: Matplotlib is a plotting library for python. This is useful for EDA and performing error analysis while training and testing the AI models.

Pillow: Pilliow is a Python Image Library is an free and open-source that supports opening, modifying, saving images. This is used to convert data sent over internet as bytes to images that can then be used in the AI models for inference. PIL is also used to load images from disk for training the models.

Colab: Google Colab is a free cloud service which helps in developing deep learning applications using popular libraries such as Keras, TensorFlow, PyTorch. In runs on Google servers with GPU or TPU acceleration which helps in reducing the training period for the AI model. Giving team, the ability to iterate and improve quickly.

Flask and Docker: Flask is a light-weight web framework for python. This can be used to run the inference server. Flask is ran inside a docker container. Docker is a set of platform as a service products that uses OS-level virtualization to deliver software in packages called containers. These container can easily be scaled up or down to automatically adjust for usage and network traffic.

Android Studio: Android Studio is the official IDE for developing native Android Apps. Android Studio provides a unified environment where you can build apps for Android phones, tablets, Android Wear, Android TV, and Android Auto. Structured code modules allow you to divide your project into units of functionality that you can independently build, test, and debug.

Android's Location library can be used to obtain user's location. Google Maps can we used to display user's location and provide additional information.

Xcode: Xcode is the official IDE for developing native iOS apps. Xcode provides a unified environment for developing native apps for all apple devices and services. Swift programming language is used to write apps in Xcode. CoreLocation framework can be used to obtain user's location. MapKit (Apple Maps) can be used to display user's location and provide additional information.

Apps can be localized to make them be available in multiple language, so more people can use the app without english being a hindrance.

Firebase: Firebase is a BaaS (Backend as a Service) provided by Google. Firebase is scalable, distributed and secure by design. It is especially geared towards business apps, with the intention of helping businesses grow their user bases and increase their profits through their mobile apps.

Firebase provides authentication(Auth), Database (Firestore), Cloud Storage (Storage), Analytics as a service.

3.2.2 Hardware Specifications

User Device Requirements:

Any Android phone running Android Oreo or above.

iPhone or iPad running iOS 13 or above.

An internet connection.

Minimum Server Requirements:

Processor: Any x86 CPU with clock speed of 2.5GHz and above.

RAM: 2GB.

Storage: At-least 10 G.B.

Above mentioned server requirements are minimum and need to be automatically scaled depending on usage and traffic.

IV ANALYSIS AND DESIGN

4.1 USE CASE DIAGRAM

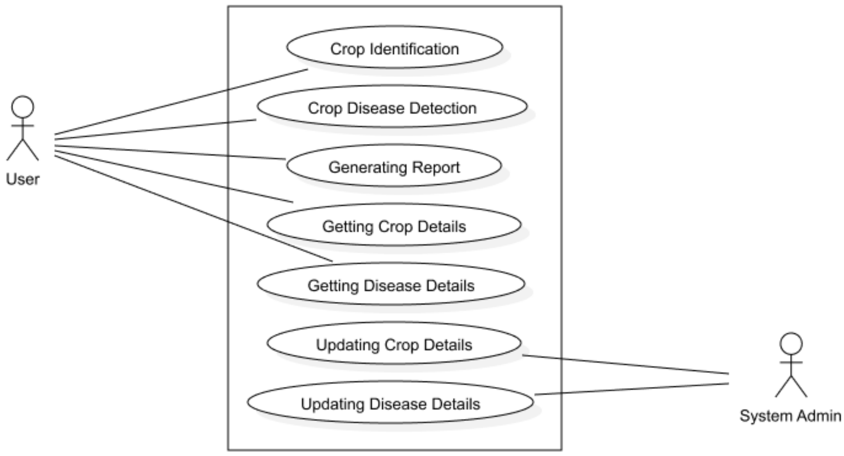


Figure 4.1: Use Case Diagram

4.2 CLASS DIAGRAMS

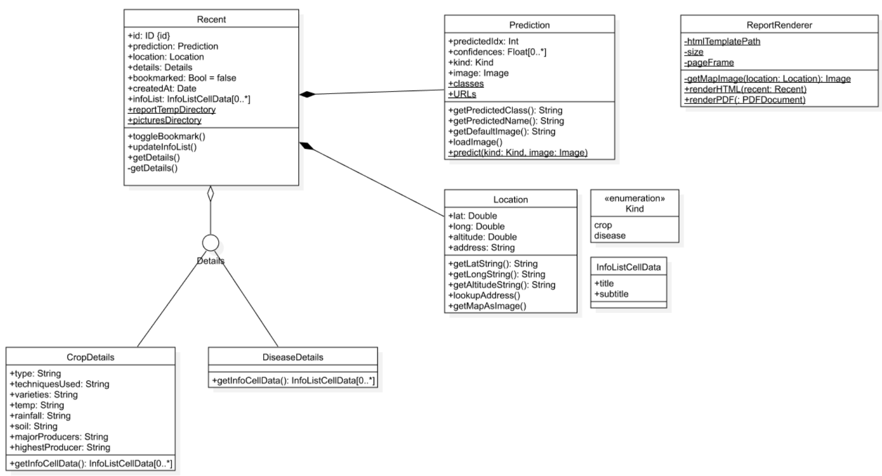


Figure 4.2: Class Diagram

4.3 ACTIVITY DIAGRAMS

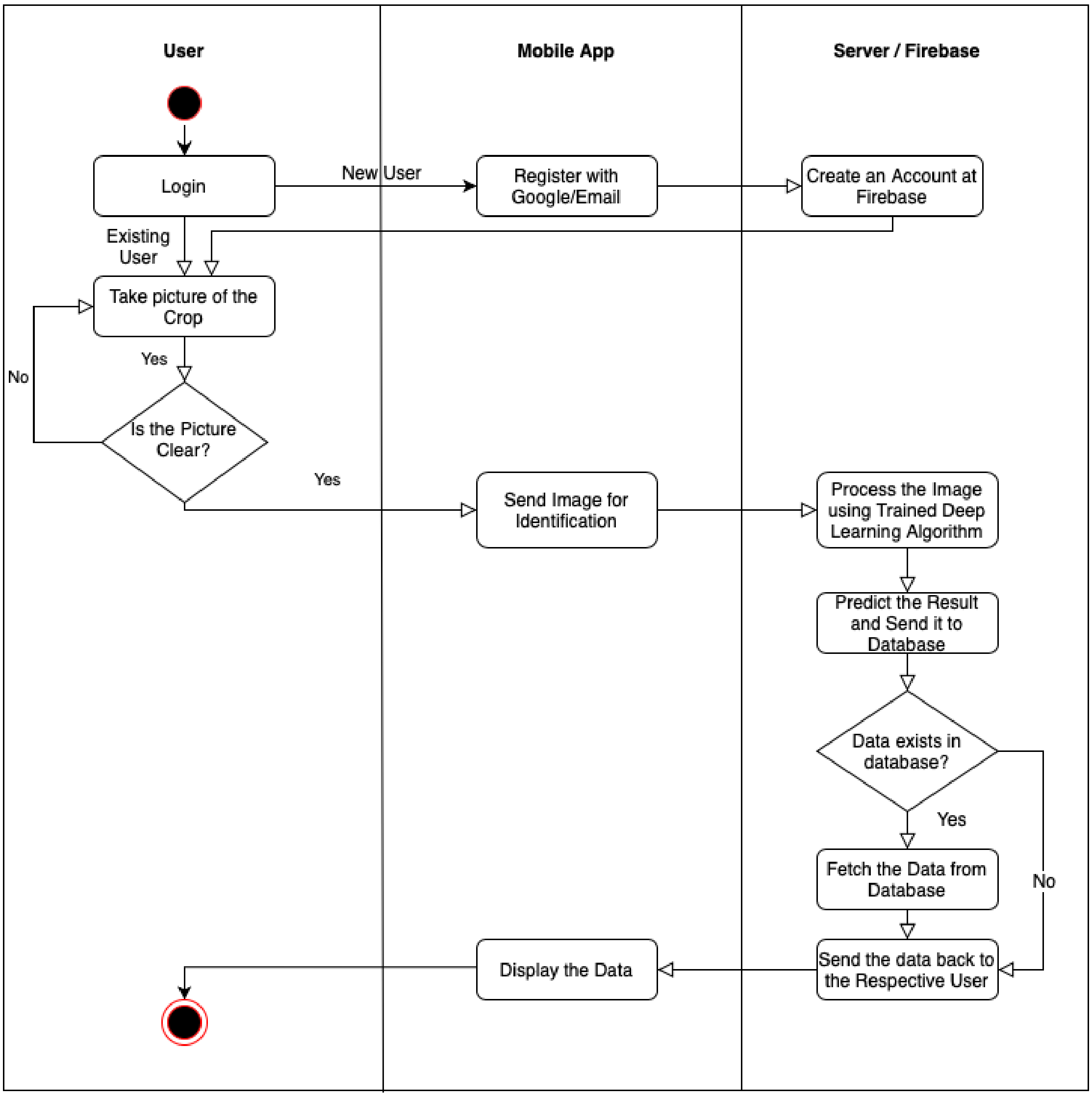


Figure 4.3: Activity Diagram for crop detection

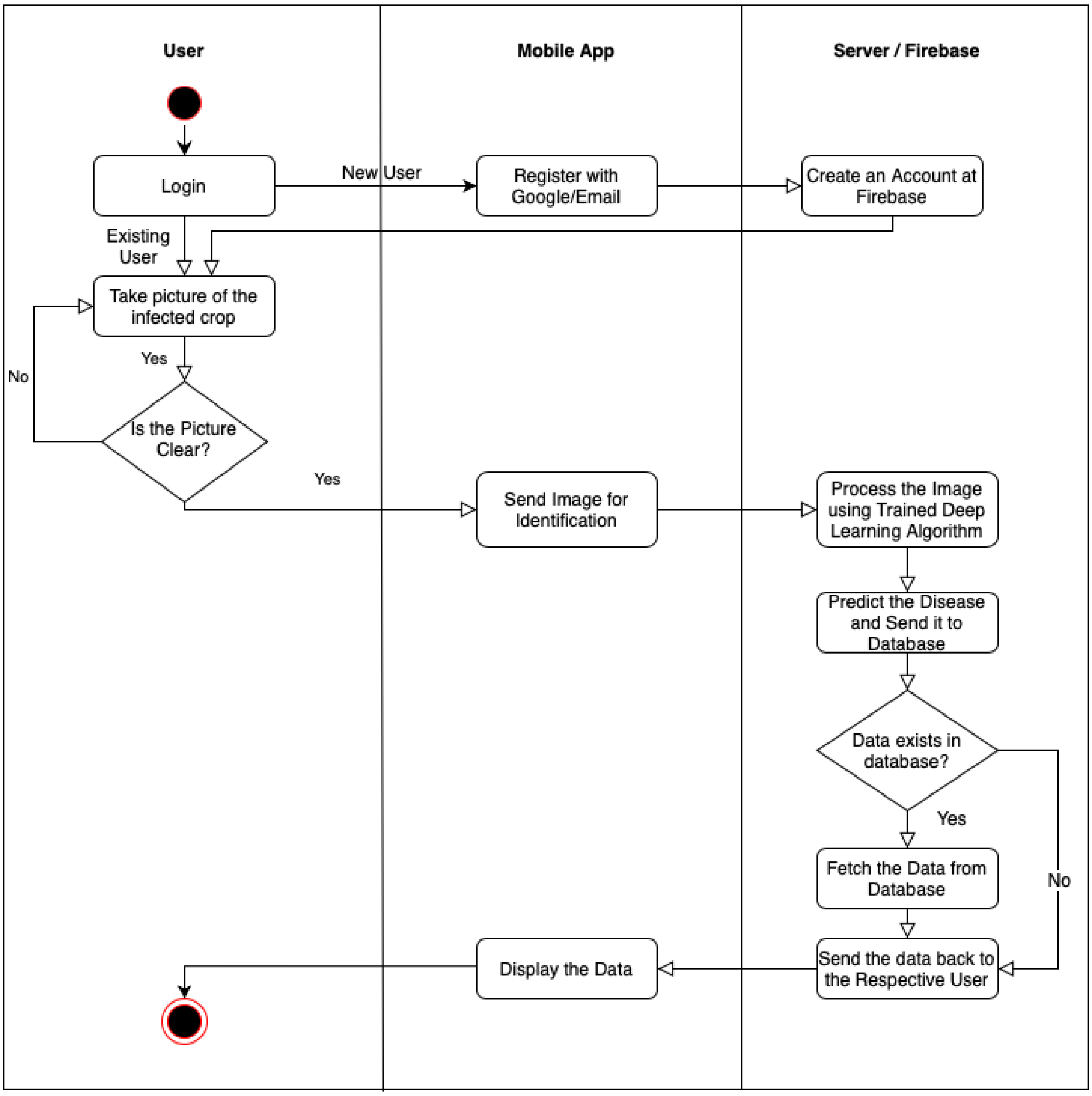


Figure 4.4: Activity Diagram for disease detection

4.4 SYSTEM ARCHITECTURE

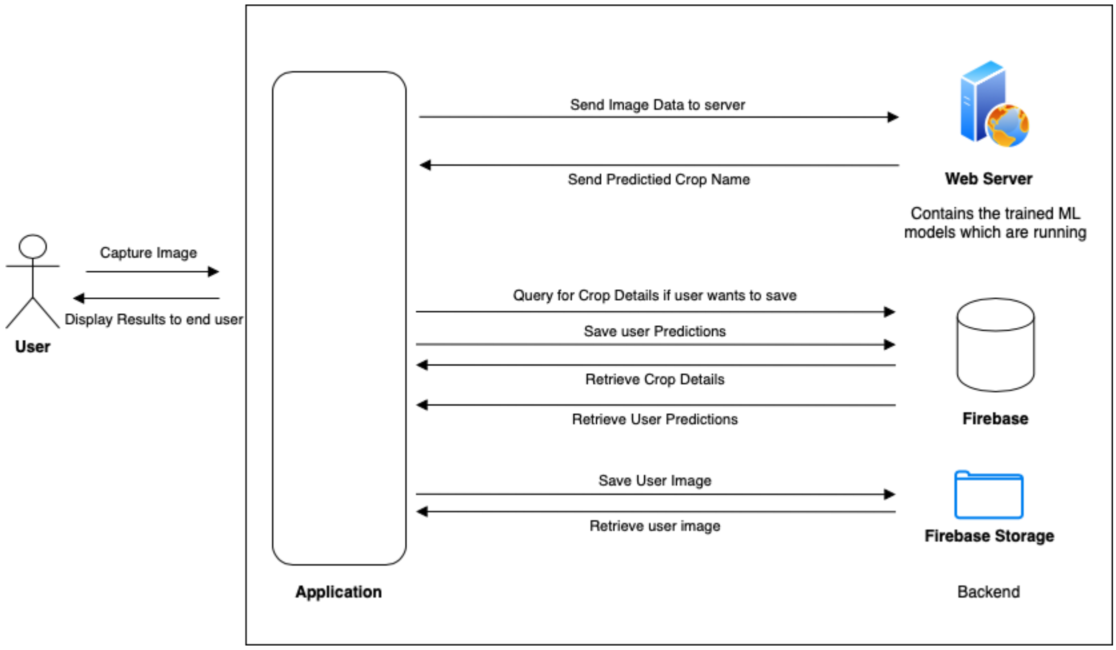


Figure 4.5: App Architecture Diagram

V IMPLEMENTATION AND TESTING

5.1 TESTING

Testing is done both manually and automatically. Manual testing involves running the app and checking if all the features are working as intended.

Automated testing involves writing test cases that evaluate the functionality of app as small individual parts and later after integration as a whole.

5.2 MOBILE APP SCREENSHOTS

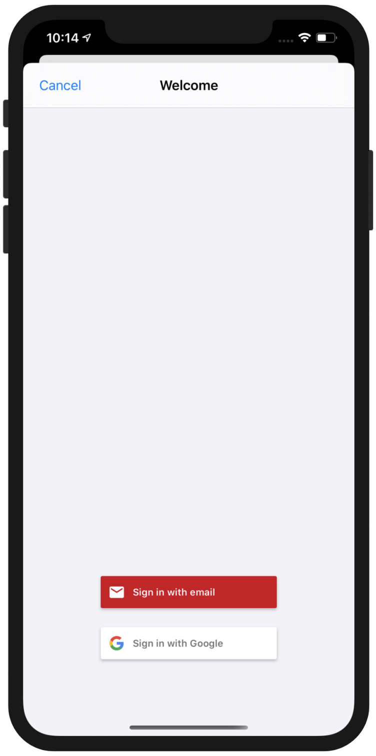


Figure 5.1: User Login

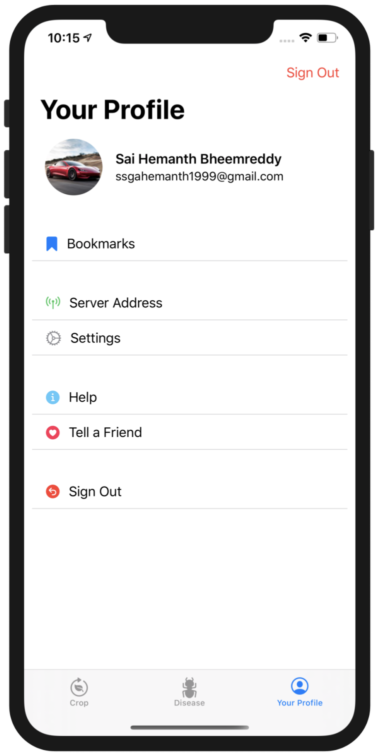


Figure 5.2: User Profile Tab

When the user first launches the app, the user is asked to login.

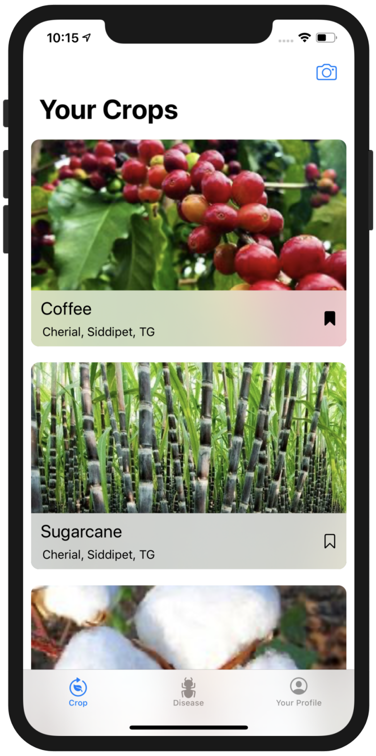


Figure 5.3: Crops Tab

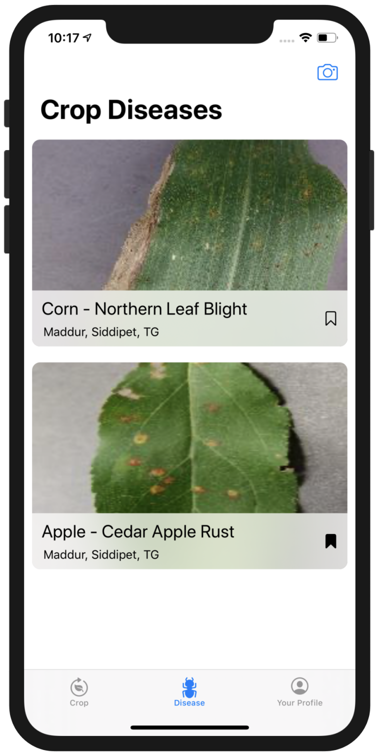
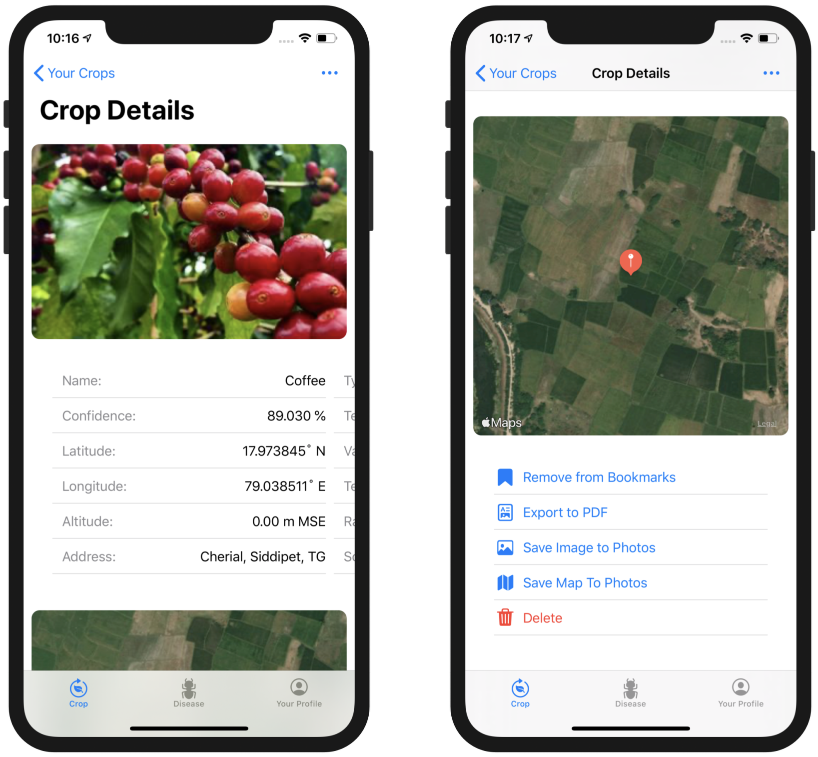


Figure 5.4: Crop Disease Tab

These tabs show user's crops and all the information is available within single tap.



(a) label 1

(b) label 2

Figure 5.5: Crop Details Screen

User can tap on any of the card in crop tab to get more details about his crops.

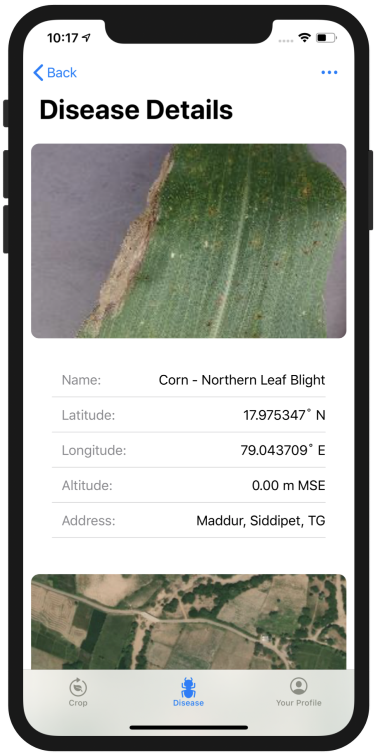


Figure 5.6: Crop Disease Screen

User can tap on any of the card in disease tab to get more details about his crop disease.

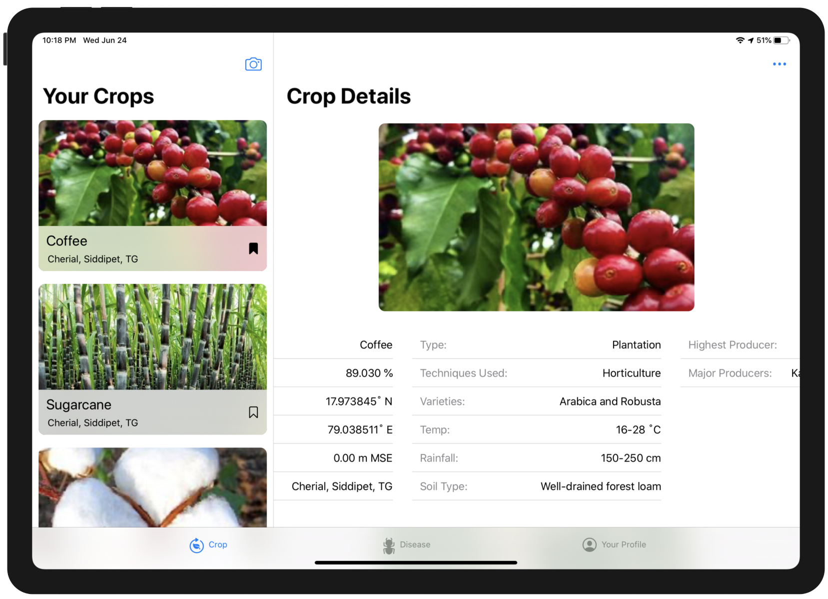


Figure 5.7: Dynamic UI

The app is made to run on any device and automatically adapts the UI to optimize the screen real estate.

VI CONCLUSION & FUTURE SCOPE

"Automation" is the frequent word being used in this decade. This project has been designed with a vision of the automation of farming, which promotes the usage of advanced machinery in agriculture. Farmer's economic growth depends on the quality of the products they produce, which relies on the plant's growth and the yield they get. With the need to move to environmental and sustainable future more than ever before, we need to find ways to produce good healthy food to feed the growing population.

We used CNN model for crop/disease detection, and have achieved an accuracy of 92% for crop detection and 90% for disease detection. But with more research in computer vision, more sophisticated model will be developed and that inevitably leads to better automation and better food production. So the short term goal is to improve the AI model, to work with more crops and increased accuracy.

The long term goal is to integrate this with an automated crop management system. This will create an end-to-end automated food production system, that can make better, healthy food with less resources, less carbon footprint and essentially zero pesticides and fertilizers. People will be able to order food on their mobile and receive freshly harvested food at their door step in less than an hour.

REFERENCES

[1] **Fukushima, Kunihiko and Miyake, Sei** *Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Visual Pattern Recognition*. Berlin, Heidelberg, 1982.

[2] **Kaiming He and Xiangyu Zhang and Shaoqing Ren and Jian Sun** *Deep Residual Learning for Image Recognition*. 2015.
`https://arxiv.org/abs/1512.03385v1`

[3] **Sharada Prasanna Mohanty and David P. Hughes and Marcel Salathé** *Using Deep Learning for Image-Based Plant Disease Detection*. 2016.
`https://arxiv.org/pdf/1604.03169`

[4] **Singh, Davinder and Jain, Naman and Jain, Pranjali and Kayal, Pratik and Kumawat, Sudhakar and Batra, Nipun** *PlantDoc: A Dataset for Visual Plant Disease Detection*. New York, NY, USA. 2020.
`https://doi.org/10.1145/3371158.3371196`

[5] **Android**
`https://developer.android.com/docs`

[6] **Docker**
`https://docs.docker.com`

[7] **Firebase**
`https://firebase.google.com/docs`

[8] **Flask**
`https://flask.palletsprojects.com/en/1.1.x/`

[9] **Google Colab**
`https://colab.research.google.com`

[10] **Google Images** *Crop Images Dataset* Collected by web scrapping images of crop from google images

[11] **iOS**
https://developer.apple.com/documentation/

[12] **Matplotlib**
https://matplotlib.org/contents.html

[13] **NumPy**
https://numpy.org/doc/stable/

[14] **Pillow**
https://pillow.readthedocs.io/en/stable/

[15] **PyTorch**
https://pytorch.org/docs/stable/index.html