**mWheat-Doctor: A Mobile based Wheat Disease Identification Application using Deep Learning Models**

**Abstract**

Computational models have been an important contributor to growth in agriculture. Artificial Intelligence (AI) has revolutionised agriculture by efficiently disseminating information to achieve food security. Plant Protection plays a significant role in achieving targets of crop production.

Agribusiness is critical to the survival of the majority of Indians, and it plays a crucial role in the finances of every nation. The disease is spread to many plants and causes huge damage. Plants are essential to the survival of all organisms on Earth. As a result, it is crucial to take steps to detect and mitigate plant diseases. Plant diseases are a major cause of crop losses in agriculture.

This project employs a Deep Learning approach to detect and classify plant diseases by examining the leaf of a given plant and classifying the disease based on image processing techniques for automated vision systems used in the agricultural field.

In this project, the image is captured in the android application where the image is then processed through the deep learning model deployed which identifies the disease in the wheat plant.

**Introduction**

In the agriculture sector, plant diseases are responsible for major economic food losses across the globe. Food losses due to crop infections from pathogens such as bacteria, viruses, and fungi are persistent issues. The situation further gets complex by the fact that, nowadays, diseases are transferred globally more easily than ever before. In order to minimise the disease induced damage in crops during growth, prevention in crops is imperative.

Computational models have been an important contributor to growth in agriculture. Artificial Intelligence (AI) has revolutionised agriculture by efficiently disseminating information to achieve food security. Plant Protection plays a significant role in achieving targets of crop production. AI has begun to modify the plant protection environment around us. AI-based equipment and machines like robots and drones have been designed for disease and weed detection. Machine Learning (ML) coupled with computer vision have potential to help farmers in protection of crops.

Traditionally, crop inspection and plant disorders were identified by farmers or experts with some training or experience. This manual method was expensive as it requires continuous monitoring and was not feasible for the larger fields. Due to complexity and variation in a large number of cultivated plant diseases, even experienced agronomists and plant pathologists fail to diagnose specific diseases accurately. It is also worth noting that many agricultural areas are too difficult to be properly monitored throughout.

Machine learning methods improve disease detection and diagnosis in comparison to conventional techniques. AI based approaches can help in disease identification at an early stages so as to focus the efforts to eliminate diseases at the identified points. Such identification will also lead to development of AI based machines which reduce human labour as well as chemical applications in disease control.

The advanced HD cameras, better computing power and communication avenues offered by smartphones offer a promising platform for automated disease detection in crops, which can save time and help in the timely management of diseases, in cases of outbreaks.Smartphones also provide an affordable and cost effective way of identification. It is easy to use and universally available.

**Table of Contents**

| S. No. | Item | Page No. |
| --- | --- | --- |
| 1 | Data and Methodology | 2-5 |
| 2 | Design | 5-10 |
| 3 | Discussion and Results | 10-11 |
| 4 | Testing | 11-13 |
| 5 | Conclusion | 13 |

**1. Data and Methodology**

**1.1 Data**

For feasibility study I made use of data provided by Google (<https://codelabs.developers.google.com/codelabs/recognize-flowers-with-tensorflow-on-android-beta/#0>) and TensorFlow (<https://www.tensorflow.org/lite/models/modify/model_maker/image_classification>).

For the purpose of mWheat-Doctor, I used the WheatNet Model developed at IASRI(Indian Agricultural Statistics Research Institute) with collaborative efforts from NIAP(National Institute of Agricultural Economics and Policy Research) and IARI(Indian Agricultural Research Institute). For this model 6500+ images have been used for identification of three wheat diseases i.e. Yellow Rust, Brown Rust and Stem Rust.

**1.2 Hardware Requirements**

| Component | Requirement |
| --- | --- |
| CPU | AMD Ryzen 5 4600h (6 Core, 3.0GHz) |
| RAM | 16GB |
| SSD | 512GB |
| GPU | NVIDIA GeForce GTX 1660Ti |

**1.3 Software Requirements**

(i) Google Colab

(ii) TensorFlow

(iii) Android Studio

(iv) Google Firebase

**1.4 Methodologies**

**Image Processing**

Image processing techniques were widely and successfully used for accurate detection and classification of the plant. Categorization is achieved as a two-dimensional taxonomy. One dimension of which specifies object recognition, data reduction/ feature extraction, pre-processing, segmentation, optimization and image understanding. In an alternate dimension, inputs are received and tasks at different levels are completed e.g. pixel level, object set level etc.

**Machine Learning**

Machine learning focuses on algorithms capable of learning on their own from a given set of input data according to the objective. Its high-performance computing creates new opportunities in the agriculture domain. Machine learning and statistical pattern recognition have been the subject of tremendous interest in the agriculture domain because they offer promise for improving the sensitivity of disease detection and diagnosis. The machine learning enabled techniques provide rich recommendations and insights for farmer decision support and action.

**Deep Learning**

With the development of computational systems, in particular, Graphical Processing Units (GPU) embedded processors, the Machine Learning-related Artificial Intelligence applications have achieved tremendous growth in recent years, leading to the development of prominent methodologies and models especially Deep Learning (DL). DL is deep neural network that refers to the addition of more depth or complexity into the model as well as transforming the data using various functions that allow data representation in a hierarchical way, through several levels of abstraction. Deep Learning (DL) is quickly becoming one of the most important tools that can be applied for solving various agricultural problems involving classification or prediction, related to computer vision and image analysis, or more generally to data analysis.

**Google Colab**

Colab is a cloud-based notebook environment that is completely free to use. It allows you and your team associates to edit documents in the same way that you do with Google Docs. Many popular machine learning libraries are supported by Colab and can be easily loaded into your notebook.

**TensorFlow**

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

**Android Studio**

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems.

**Kotlin**

Kotlin is a cross-platform, statically typed, general-purpose programming language with type inference. Kotlin is designed to interoperate fully with Java, and the JVM version of Kotlin's standard library depends on the Java Class Library, but type inference allows its syntax to be more concise.

**Flask API**

Flask is a widely used micro web framework for creating APIs in Python. It is a simple yet powerful web framework which is designed to get started quickly and easily, with the ability to scale up to complex applications.Since it is a micro-framework, it is very easy to use and lacks most of the advanced functionality which is found in a full-fledged framework.

**Cloud Storage**

Cloud storage is a cloud computing model that stores data on the Internet through a cloud computing provider who manages and operates data storage as a service. It's delivered on demand with just-in-time capacity and costs, and eliminates buying and managing your own data storage infrastructure.

**Google Firebase**

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

**2. Design**

**2.1 User Interface**

**Landing Screen**

A landing page is a standalone web page that a person "lands" on after clicking through from an email, ad, or other digital location. Once they're on your landing page, users are encouraged to take an action, such as joining your list or buying your products. mWheat-Doctor’s landing page is also the login page for the app.

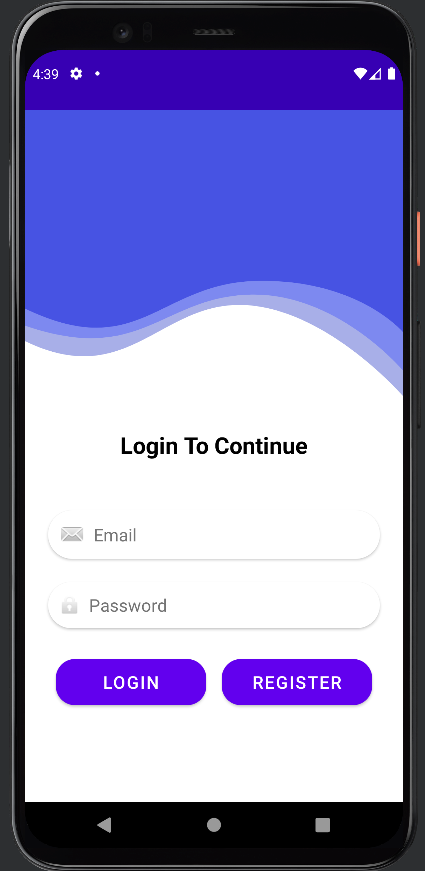


Fig 1: Landing Screen

**Registration Screen**

Screen where a user enters their personal information and registers to a system. Here the user entered details are stored in a cloud network in Google Firebase.

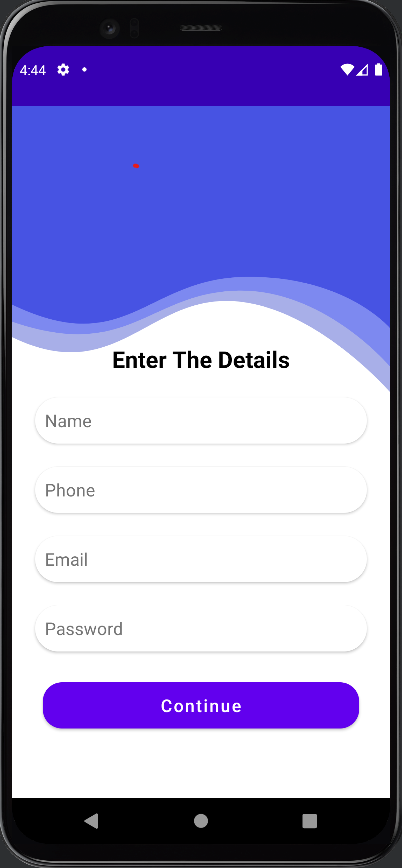


Fig 2: Registration Screen

**Information Screen**

All the details entered while registration are shown in this screen from Google Firebase. An option for continuing with the app and logging out is provided.

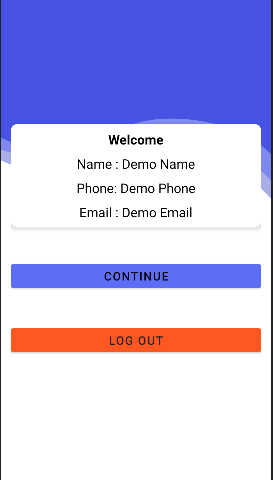


Fig 3: Information Screen

**Image Capturing Screen**

Here the user is asked to click a picture or upload a picture of the plant they want to check for the disease.

It then also processes the image and provides the output

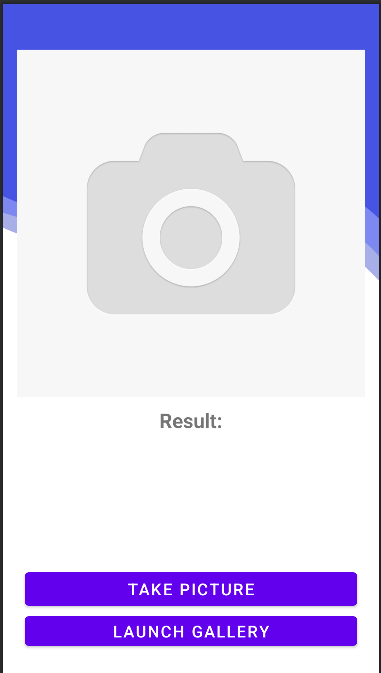
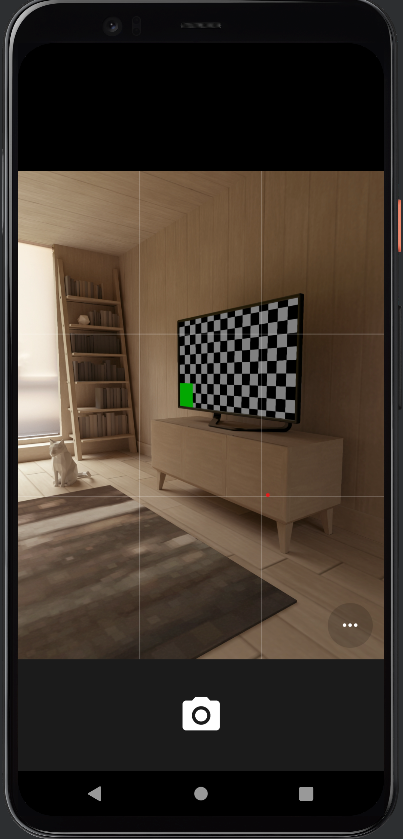
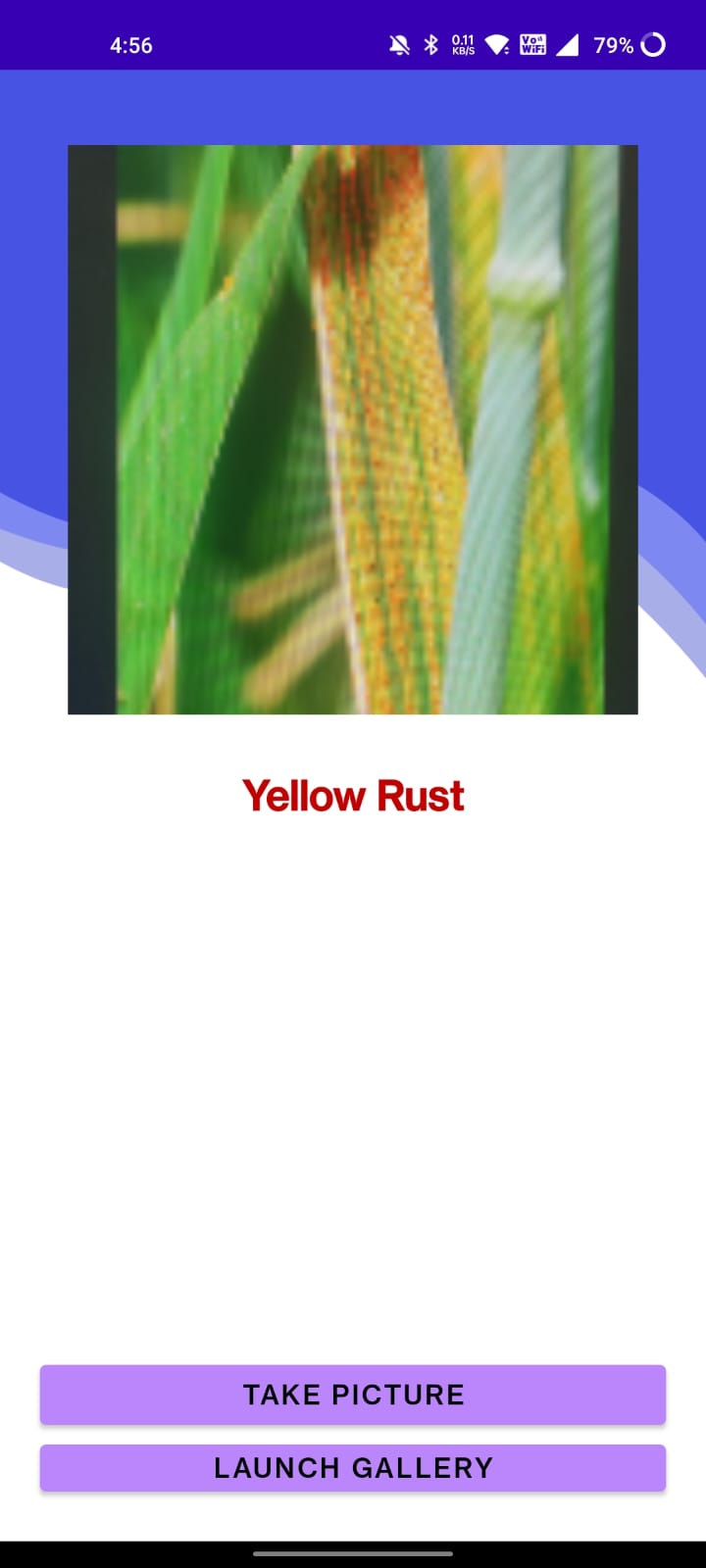
  

Fig 4: Interface Fig 5: Image Capture Screen Fig 6: Result

**2.2 Backend**

**Login Database using Google FireBase**

Firebase Authentication aims to make building secure authentication systems easy, while improving the sign-in and onboarding experience for end users. It provides an end-to-end identity solution, supporting email and password accounts, phone auth, and Google, Twitter, Facebook, and GitHub login, and more.

FirebaseUI provides a customizable, open source, drop-in auth solution that handles the UI flows for signing in users. The FirebaseUI Auth component implements best practices for authentication on mobile devices and websites, which can maximise sign-in and sign-up conversion for your app.

**Firestore Database**

Cloud Firestore is a NoSQL document database that lets you easily store, sync, and query data for your mobile and web apps - at global scale.

Cloud Firestore ships with mobile and web SDKs and a comprehensive set of security rules so you can access your database without needing to stand up your own server. Using Cloud Functions, our serverless compute product, you can execute hosted backend code that responds to data changes in your database. Of course, you can also access Cloud Firestore with traditional client libraries too (i.e. Node, Python, Go, and Java).

**Deploying AI Model**

Building a REST API allows you to use your model to make predictions for different clients. Almost any device can speak REST - Android, iOS, Web browsers, and many others.

Flask allows you to build a REST API in just a couple of lines. *Of course, we’re talking about a quick-and-dirty prototype*.

The API has a single route (index) that accepts only POST requests. *Note that we pre-load the data transformer and the model*.

The request handler obtains the JSON data and converts it into a Pandas DataFrame. Next, we use the transformer to pre-process the data and get a prediction from our model. We invert the log operation we did in the pre-processing step and return the predicted price as JSON.

We’ll deploy the project to Google App Engine.

App Engine enables developers to stay more productive and agile by supporting popular development languages and a wide range of developer tools.

**2.3 Steps involved in Making mWheat-Doctor**

(i) We start with making the user interface in Android Studio. We make all the required screens which are present in .xml files in Android Studio.

(ii) After completing the user interface we move on to coding where we connect all the buttons and screens together which we made in the user interface using Kotlin.

We connect various buttons to different screens using the setOnClickListener function.

(iii) Now that all the screens are connected, we ask the application for various required permissions that will be required by the user to perform the functions of the application i.e. Camera Permissions and Storage Permissions.

(iv) Now we connect the google firebase and firestore database with the application.

(v) Now we deploy the AI Model which we made through the Flask API. We capture the image in the same pixel size as trained in the model which 380\*380.

(vi) From the model we take the output as Result which is displayed in the Result Screen.

**3. Discussion and Results**

**3.1 Discussion**

Before making the mWheat-Doctor app, I made a Flower Recognition app with a tensorflow lite AI Model for checking the feasibility of the project.

By doing this I learned to make a .tflite AI Model using Google Colab where I trained 3670 photos of five different flowers i.e. Daisy, Rose, Sunflower, Dandelion and Tulips in a sequential model.

After getting the trained model I deployed it in an application made using android studio.

From this I learned that the project planned was feasible and moved forward with its development.



Fig 7: Flower Recognition App

**3.2 Result**

**Functionality of mWheat-Doctor**

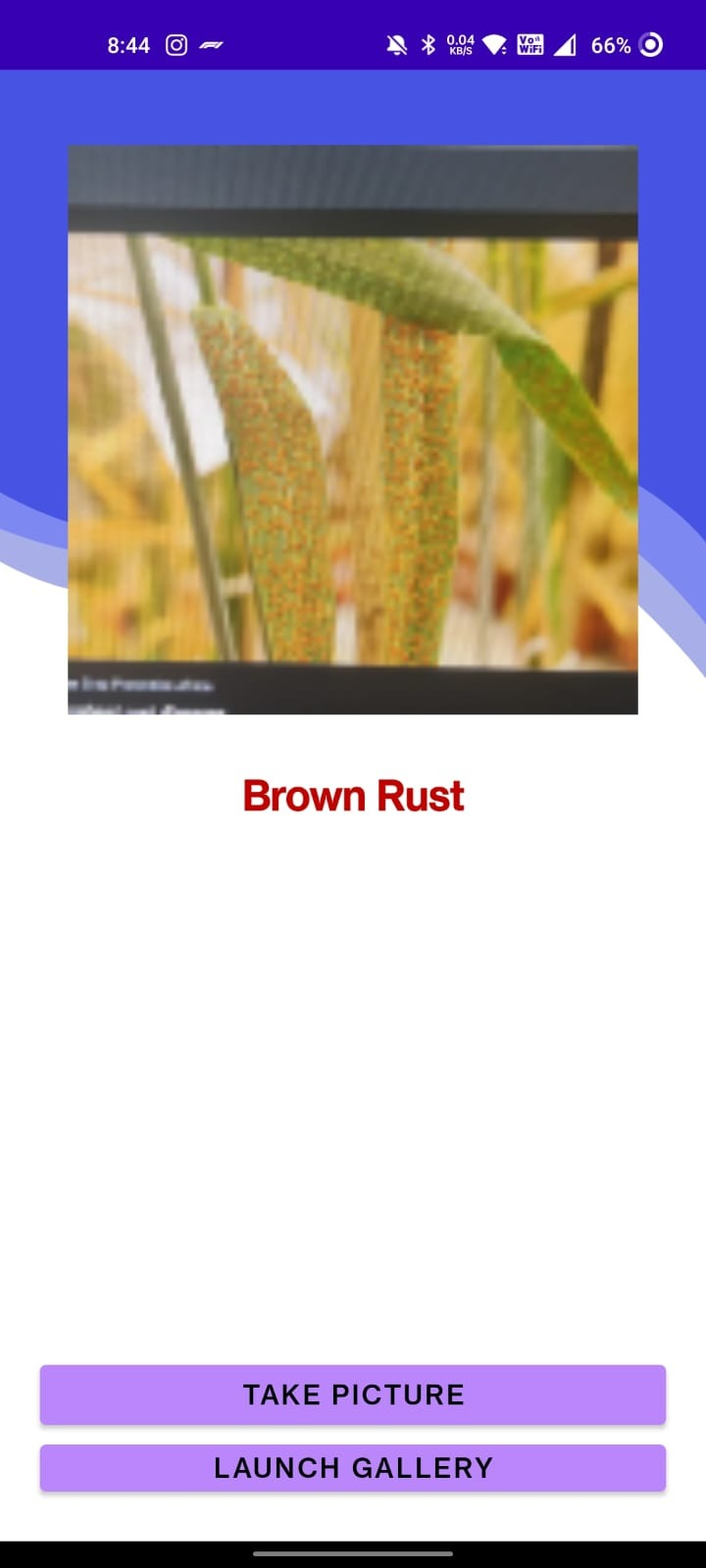
First the user has to register to mWheat-Doctor. After registering the user enters their credentials on the login screening. Then they are taken on to the next screen where their details are confirmed and then they continue to the disease identification section. Here the user gets two options, either to click the image of the plant or to upload the image from the gallery.

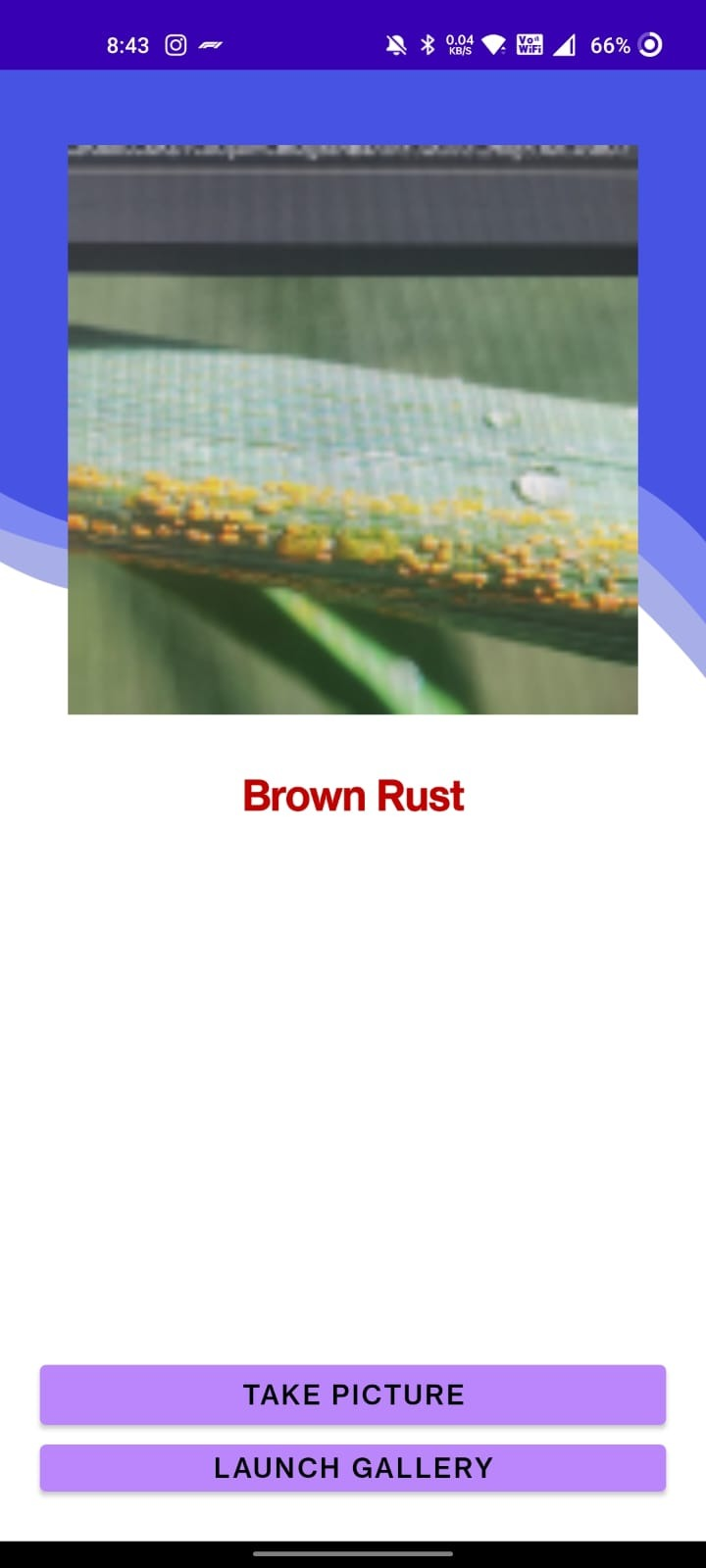
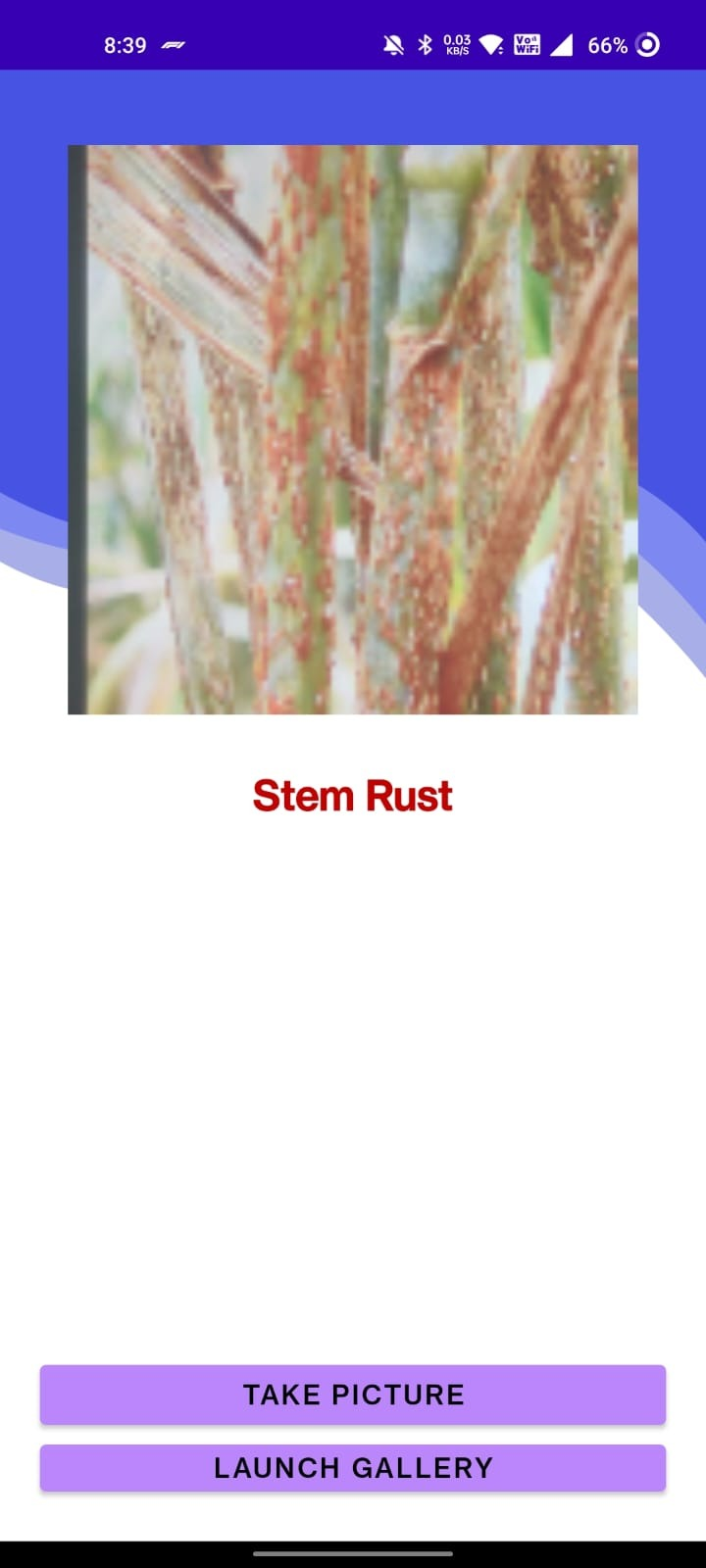
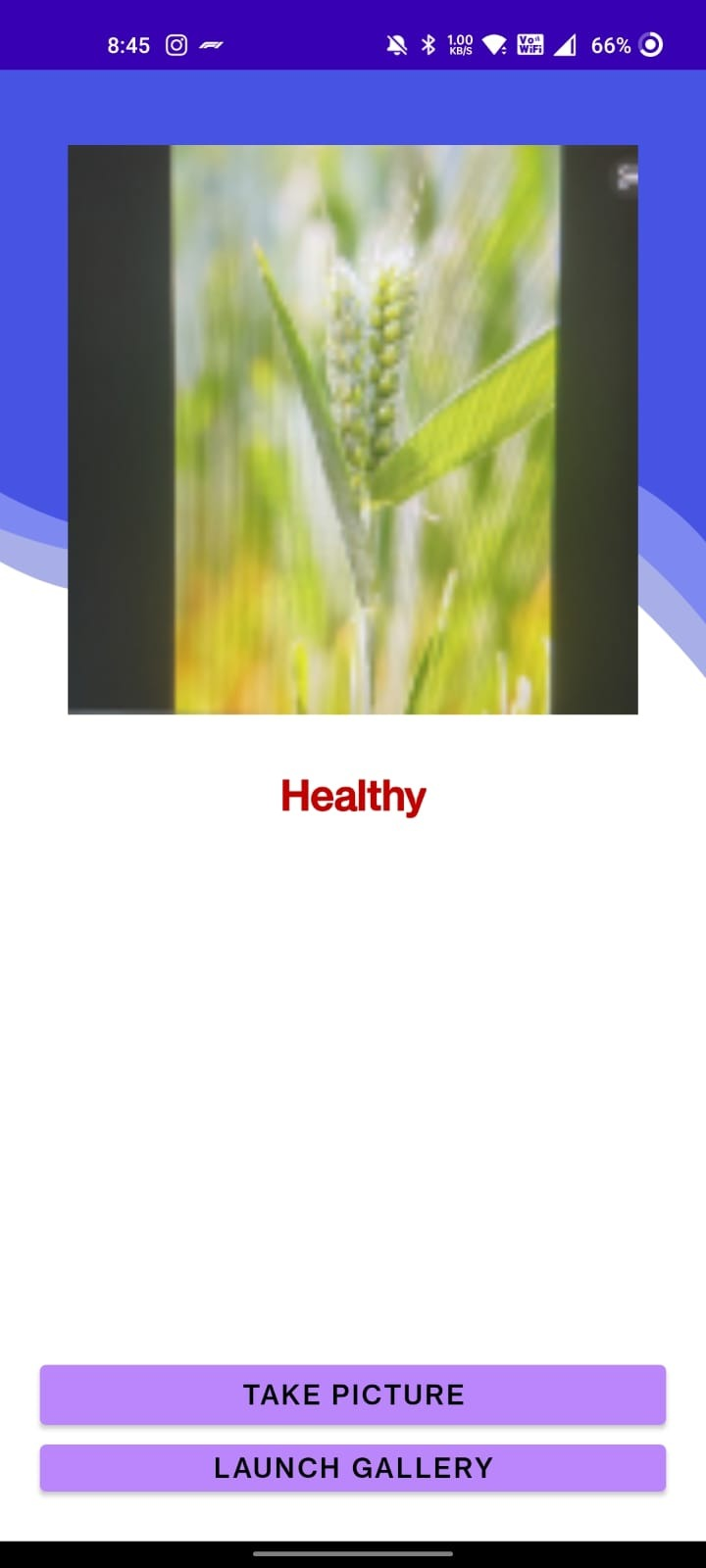
After that the AI Model runs in the background and gives the user the necessary result.

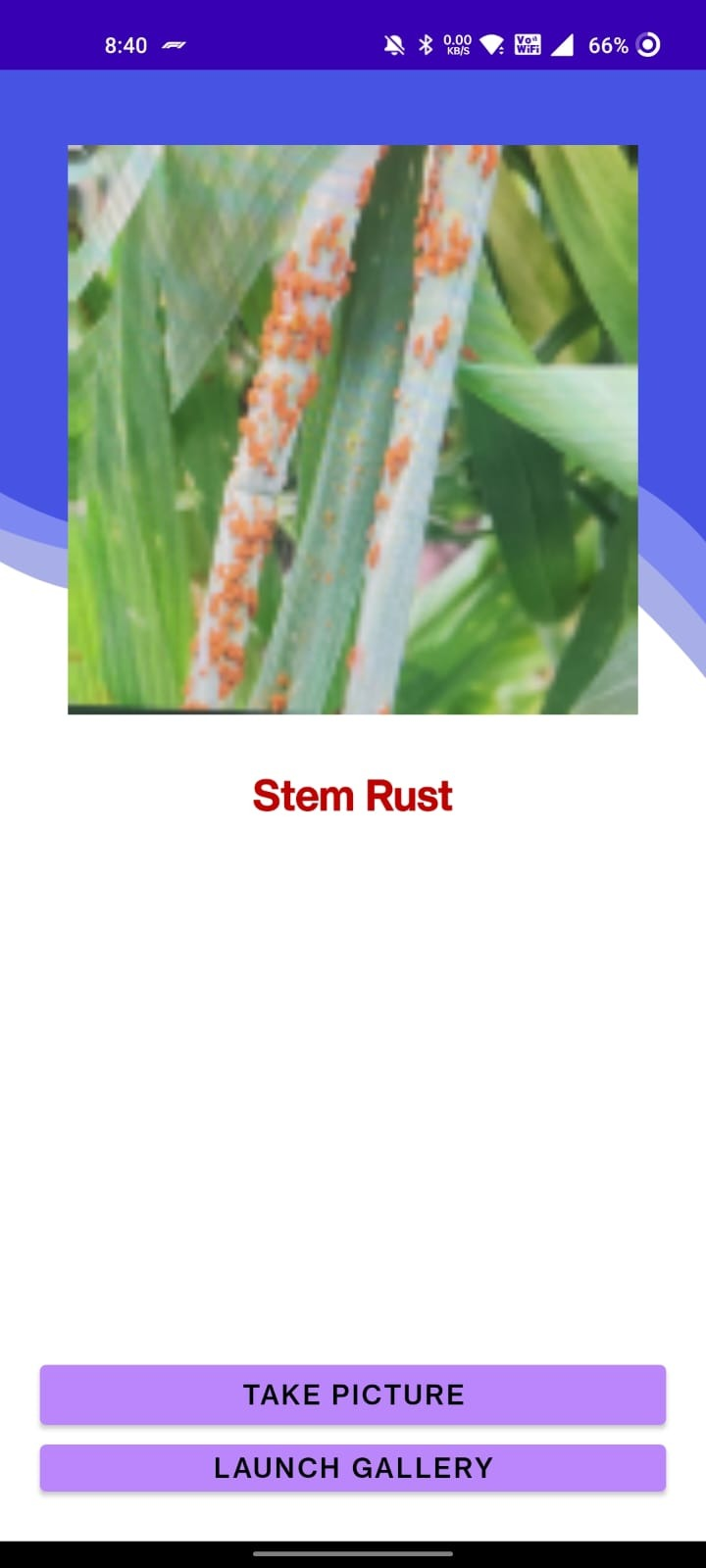
If the user is already registered they need not register again and can simply continue to the disease identification section.

**4. Testing**

Now we test the application with 10 leaves.

****

****

****

**Fig 8-17: Testing of the App**

**5. Conclusion**

Finally, it can be stated that Plant Disease Detection is playing an increasingly important role in shaping the future of farming. Protecting crops on a farm is a difficult task that requires extensive knowledge of the crop being grown and potential pests. The developed application addresses the issue of manual detection of non-real-time plant diseases.

This project explores the potential of artificial intelligence based algorithms for disease identification. The implementation of artificial intelligence based approaches to plant protection and specifically on leaf image classification and plant disease identification has been presented.

The effective and easy use of Android Application is shown in this project. Implementing this can help reduce disease spread at a rapid extent.