AI-Powered Crop Disease Prediction and Agricultural Solutions: A Comprehensive Framework for Enhanced Crop Health Management

Task - 0

Adarsh Kumar

21st December, 2023

Abstract

In response to the escalating challenges posed by crop diseases in agriculture, this report introduces an innovative mobile application, titled "AI-Powered Crop Disease Prediction and Agricultural Solutions." The app leverages cutting-edge machine learning algorithms, including support vector machines (SVM) and convolutional neural networks (CNN), to provide farmers with a holistic solution for disease diagnosis, personalized treatment recommendations, product procurement, and expert consultations. The comprehensive business model encompasses revenue streams from fertilizer sales, expert consultations, and a freemium subscription model, ensuring sustainability and scalability.

1. Problem Statement

Crop diseases pose a significant threat to farmers' livelihoods, leading to reduced crop yields, increased production costs, and lower profits. Traditional methods of disease diagnosis and treatment are often time-consuming, inaccurate, and lack the necessary expertise. This results in substantial economic losses and food security risks The need for a cost-effective, AI-driven solution to enhance disease prediction, recommend tailored treatments, and facilitate easy access to agricultural products is critical.

2. Market/Customer/Business Need Assessment

The global agricultural market is vast and growing, with a significant demand for innovative solutions to address crop diseases. The agriculture market worldwide is projected to grow by 5.66% (2023-2028) resulting in a market volume of US$4.86tn in 2028. And the global fertilizer market amounted to more than 193 billion U.S. dollars in 2021, an increase of roughly 12 percent in comparison with the previous year. It is forecast that the fertilizer market will surpass 240 billion U.S. dollars by 2030.

Farmers are eagerly seeking technologies that can help them identify and manage crop diseases efficiently and cost-effectively. The market for crop disease prediction apps is rapidly expanding, driven by the increasing adoption of smartphones and the growing awareness of digital technologies in agriculture.

3. Target Specifications and Characterization

The crop disease prediction app aims to provide farmers with a user-friendly and accurate tool to detect and manage crop diseases. The app will offer the following features:

- **Image-based disease diagnosis:** Farmers can capture images of affected plant parts and upload them to the app. The app will use machine learning algorithms to analyze the images and identify the disease with high accuracy
- **Treatment recommendations:** Based on the disease diagnosis, the app will provide personalized treatment recommendations, including recommended fertilizers, chemicals, or organic matter.
- **Product delivery:** Farmers can order the recommended agricultural products directly from the app. The app will aggregate products from multiple fertilizer companies and deliver them to farmers through third-party logistics providers.
- **Expert consultation:** Farmers can book consultations with agricultural experts through the app. The experts will provide guidance on disease management, crop care, and other agricultural practices.

4. External Search

Research has been conducted to identify existing technologies and best practices related to crop disease prediction, machine learning, image processing, and mobile app development. Several open-source frameworks and APIs are available for developing such an app. Additionally, numerous academic papers and articles provide valuable insights into the state-of-the-art in crop disease prediction. Some of these which I referred were:

- https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10086516
- (PDF) Leaf Disease Detection Using Machine Learning (researchgate.net)
- (PDF) Leaf Disease Detection using Image Processing (researchgate.net)
- https://www.statista.com/outlook/io/agriculture/worldwide

5. Benchmarking Alternate Products

Several existing crop disease prediction apps have been evaluated. These apps offer varying features, accuracy levels, and user experiences. Some of the key competitors include Plantix, Farmkey, AgriApp, and Agrio. The proposed app aims to differentiate itself through its superior accuracy, comprehensive treatment recommendations, direct product delivery, and expert consultation services.

COMPETITORS	FEATURES				
	Disease Detection	Treatment Recommendations	Direct Product Delivery	Expert Consultation	Fertilizer Calculator for Crop & Plot
Plantix	✓	✓	✗	✓	✓
Farmkey	✗	✗	✓	✗	✗
AgriApp	✗	✗	✓	✓	✗
Agrio	✓	✓	✗	✗	✗
Our App	✓	✓	✓	✓	✗

6. Applicable Patents

A thorough search for relevant patents can be conducted. So far, these patents could be identified:

- Image-based disease diagnosis
- ML algorithms
- Mobile app platforms

The proposed app will ensure compliance with all applicable patents and intellectual property rights.

7. Applicable Regulations

The app will comply with all relevant regulations governing data privacy, product quality and safety, intellectual property rights, and tax and trade policies. The app will also adhere to ethical guidelines and best practices in data handling and use.

8. Applicable Constraints

- **Data Availability:** The availability of high-quality data on crop diseases and treatment options is a challenge. We will address this challenge by collaborating with agricultural organizations and research institutions to collect and curate data.
- **Technical Expertise:** The development and deployment of the app require technical expertise in ML, image processing, and mobile app development. We will address this challenge by partnering with other experts in these fields, and self-learning as much as possible.
- **Regulatory Hurdles:** The app must comply with relevant regulations governing data privacy, product quality and safety, and intellectual property rights. We will address this

challenge by conducting thorough legal and regulatory research and obtaining the necessary approvals.

9. Business Model

We can generate revenue through multiple streams:

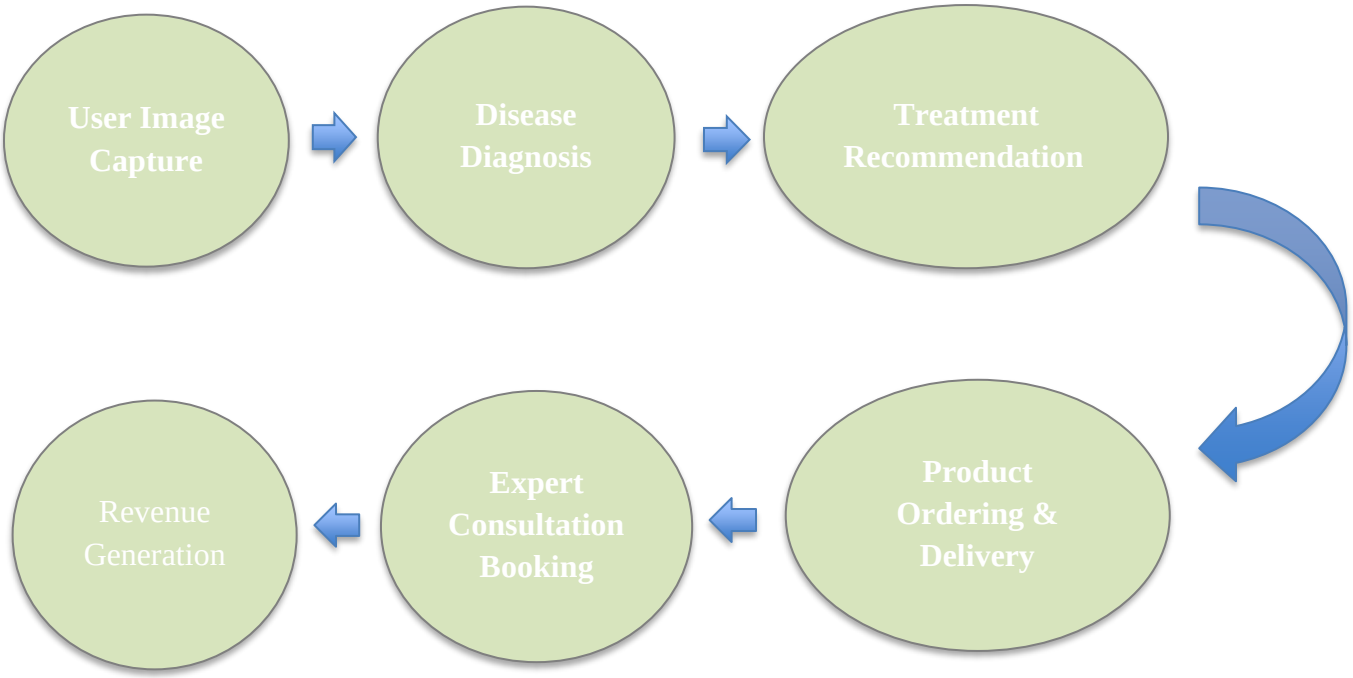
- **Margin and commission from fertilizer companies:** The app will earn a margin on the sale of fertilizers and other agricultural products. Additionally, it will receive a commission from fertilizer companies for promoting their products.
- **Consultation fee with experts:** Farmers will pay a consultation fee to access the expertise of agricultural experts through the app.
- **Freemium model:** The app will offer a limited number of free predictions to farmers. For additional predictions and access to premium features, farmers will need to subscribe to a paid plan.

10. Concept Generation

As a farmer's son from a village where crop diseases wreak havoc on livelihoods, I recognized the urgent need for a comprehensive solution to combat this challenge. I envisioned an app that would empower farmers with the knowledge and resources to effectively manage crop diseases. Through brainstorming and research, I conceptualized a user-friendly app that seamlessly integrates crop disease diagnosis, personalized treatment recommendations, direct product delivery, and access to agricultural experts. This app would leverage cutting-edge technology to provide accurate and timely information to farmers, enabling them to make informed decisions and protect their crops.

11. Final Product Prototype and Product Details

Schematic Diagram:



Overall Architecture:

- **User Interface:** The user interface (UI) is the part of the app that the user interacts with directly. It includes features such as a camera for taking pictures of affected plants, a form for entering data about the crop and affected area, and a display for showing the disease diagnosis and treatment recommendations.
- **Mobile App Development Framework:** The front end of the app can be developed using a mobile app development framework such as React Native or Flutter. These frameworks allow us to build cross-platform apps that can be deployed to both iOS and Android devices.
- **Back End:** The back end of the app is responsible for processing user requests, storing data, and generating responses. It includes components such as a web server, database, ML model, and API.
- **Data Sources:** The app utilizes various data sources, including images of crop diseases, treatment recommendations, and expert advice. These data sources are used to train the ML model and provide personalized recommendations to farmers.
- **Algorithms:** The app employs various ML algorithms like SVM and CNN for disease prediction and treatment recommendation. These algorithms are trained on large datasets of crop diseases and treatment information.
- **Frameworks and Software:** The app can be developed using a variety of frameworks and software, including Python, Keras, TensorFlow, and React Native. These tools and technologies enable the development of robust and user-friendly app features.

12. Code Implementation

- This is sample code without actual data which uses machine learning techniques to identify and treat diseases based on the symptoms and the image of the affected crop.
- It uses two different models to predict the disease: a support vector machine (SVM) that compares the symptoms with the descriptions of the diseases, and a convolutional neural network (CNN) that analyzes the image of the crop.
- It also uses a Jaccard similarity measure to recommend the most suitable treatments for the predicted disease, based on the similarity between the treatments and the diseases.
- Github Link

```python
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import jaccard_score
from keras.models import load_model
import cv2
import matplotlib.pyplot as plt

#Loading the data
data = pd.read_csv('data.csv')

#Creating TF-IDF vectorizers for symptoms and treatments
symptoms_vectorizer = TfidfVectorizer()
treatments_vectorizer = TfidfVectorizer()

#Converting the symptoms and treatments to numerical vectors
symptoms_vectors = symptoms_vectorizer.fit_transform(data['symptoms'])
treatments_vectors = treatments_vectorizer.fit_transform(data['treatments'])

#Creating a numerical vector for each disease using the symptoms vectorizer
disease_vectors = symptoms_vectorizer.transform(data['disease_name'])

#Loading the CNN model
model = load_model('cnn_model.h5')

#Defining a function to predict the disease using SVM
def predict_disease_svm(symptoms):
    #Converting the symptoms to a numerical vector
    symptoms_vector = symptoms_vectorizer.transform([symptoms])
    #Calculating the cosine similarity between the symptoms vector and the disease vectors
    similarity_scores = symptoms_vector @ disease_vectors.T #Using matrix multiplication instead of cosine_similarity

    #Getting the index of the most similar disease
    most_similar_disease_index = np.argmax(similarity_scores)

    #Returning the name of the most similar disease
    return data['disease_name'][most_similar_disease_index]

#Defining a function to predict the disease using CNN
def predict_disease_cnn(image):
    #Preprocessing the image
    image = cv2.resize(image, (224, 224))
    image = np.array(image) / 255.0

    #Predicting the disease using the CNN model
    prediction = model.predict(np.expand_dims(image, axis=0))

    #Returning the name of the predicted disease
    return data['disease_name'][np.argmax(prediction)]

#Defining a function to recommend treatments
def recommend_treatments(disease):
    #Getting the list of recommended treatments for the disease
    treatments = data[data['disease_name'] == disease]['treatment'].tolist()

    #Converting the treatments to numerical vectors
    treatment_vectors = treatments_vectorizer.transform(treatments)

    #Calculating the Jaccard similarity between the treatment vectors and the disease vector
    similarity_scores = np.apply_along_axis(jaccard_score, 1, treatment_vectors.toarray(), disease_vectors.toarray()) #
```

```
    #Getting the indices of the most similar treatments
    most_similar_treatment_indices = np.argpartition(similarity_scores, -5)[-5:]

    #Returning the list of most similar treatments
    return np.take_along_axis(data['treatment'], most_similar_treatment_indices, axis=0)

#User input
symptoms = input("Enter the symptoms of the crop disease: ")
image_path = input("Enter the path to the image of the affected crop: ")

#Predicting the disease using SVM and CNN
predicted_disease_svm = predict_disease_svm(symptoms)
predicted_disease_cnn = predict_disease_cnn(cv2.imread(image_path))

#Printing the predicted diseases
print("Predicted disease using SVM:", predicted_disease_svm)
print("Predicted disease using CNN:", predicted_disease_cnn)

#Recommending treatments for the predicted disease
treatments = recommend_treatments(predicted_disease_cnn)

#Printing the recommended treatments
print("Recommended treatments:")
for treatment in treatments:
    print("-", treatment)

#Displaying the image to the user
plt.imshow(cv2.imread(image_path))
plt.show()

#Displaying the model architecture to the user
model.summary()
```

13. Conclusion

The crop disease prediction app offers a valuable solution to address the challenges of crop diseases faced by farmers. The app's accurate disease diagnosis, personalized treatment recommendations, direct product delivery, and expert consultation services will empower farmers to effectively manage crop diseases and improve their yields. The app's business model is sustainable and scalable, with multiple revenue streams.

14. Recommendations for Future Improvements

- Expand the app's capabilities to include additional crops and diseases.
- Include fertilizer quantity calculator for crop and plot size as available in Plantix
- Integrate advanced ML algorithms to improve the accuracy of disease diagnosis and treatment recommendations.
- Develop a mobile app version of the app to make it more accessible to farmers.
- Partner with agricultural organizations and extension services to promote the app and provide support to farmers.