Feynn Labs Consultancy Services

INTELLIGENT CROP DISEASE PREDICTION

BY

Contributors: Kancharla.Sindhupriya, Shreyas Gosavi, Sanchit Singla, AkshayLanjewar, Hiya Nachnani.

Feasibility:

ML and DL techniques, like CNNs, are being successfully used for detecting plant diseases by recognizing patterns in images of plants. For this to work, we need a lot of pictures showing healthy plants and those with different diseases. There are now many online databases where we can find these images, and people are also helping by contributing to these collections. However, running these models requires powerful computers with special hardware like GPUs, and they also need to be set up on cloud platforms for them to work in real-time. So, to make plant disease detection with ML and DL feasible, we need good technology, lots of labeled images, and the right hardware and software setup.

Viability:

ML and DL-based plant disease detection to be viable, the systems need to accurately and reliably identify diseases, even in different environments and varying image qualities. This accuracy ensures that farmers can trust the system's recommendations for managing diseases effectively. Additionally, the system should be scalable to handle large deployments across different agricultural settings and adaptable to changes in disease patterns and crop varieties over time. User-friendly interfaces, such as easy-to-use dashboards and mobile apps, are crucial for farmers and agronomists to access and interact with the system effortlessly, enhancing its overall viability and usability.

Monetization:

To make money from plant disease detection services, there are a few ways to do it. One way is to charge farmers a regular fee for using the service, kind of like a subscription. This fee would give them access to all the features, like disease identification and advice on how to deal with the problems. Another option is to charge farmers each time they use the service, like paying for each analysis or consultation. You could also offer extra-special features and support for a higher fee, giving farmers more advanced disease analysis and personalized advice. Lastly, by collecting and analyzing all the data from the plant disease lidetection systems, there's an opportunity to make money by sharing valuable insights with researchers, businesses, and government agencies who are interested in using the information for things like studying agriculture trends and protecting crops.

1. Problem Statement

Crop Disease are conditions that adversely effect health and productivity of plants cultivated for food and other purposes. These diseases can lead to losses, food security and other effects. The idea is to develop an intelligent crop disease prediction system that utilizes advanced machine learning algorithms to accurately identify and predict potential diseases affecting crops based on input data such as images, environmental factors, and historical disease patterns.

2. Customer Needs Assessment

The agricultural industry is facing significant challenges, which have led to a market and corporate demand for an Intelligent Crop Disease Prediction system. Farmers around the world struggle to identify diseases in a timely manner, which results in large crop losses. This project uses machine learning to meet the market need for an effective and proactive solution. A dependable tool is necessary to improve crop management techniques, reduce losses, and maximize production for important stakeholders, such as farmers, agricultural businesses, and organizations. The target market is made up of people from a variety of geographical areas where agriculture is an important economic sector. By addressing this need, the initiative contributes to the increasing need in contemporary agriculture for technologically advanced, sustainable solutions.

3. Target Specifications and Characterization

End users like small and large-scale farmers, agricultural enterprises, and initially targeting regions with significant agricultural dependence. Major crops such as wheat, rice, corn, and soybeans are prioritized. The system should seamlessly integrate with existing agricultural technologies and demonstrate scalability for larger datasets and diverse crops. Accessibility through user-friendly interfaces and mobile applications is crucial. Collaboration with agricultural services, research institutions, and governments enhances the system's impact. Robust data privacy, security measures, and compliance with regulations are prioritized. Cost-effective pricing models accommodate different users. Comprehensive training, tutorials, and customer support ensure effective adoption. Benchmarks for accuracy, precision, speed, and efficiency are set against industry standards, and a positive user experience is emphasized. A feedback mechanism facilitates continuous improvement, adapting the system to evolving user needs and technological advancements, making it a reliable tool for proactive crop disease management.

4. Benchmarking

1. *PlantVillage's Nuru:*

An AI-based technology for agricultural disease diagnosis called Nuru was created by the non-profit company PlantVillage. Nuru detects illnesses in a variety of crops by using machine learning algorithms that have been trained on large datasets.

- Accuracy and Precision: Nuru outperformed conventional techniques in identifying a variety of crop diseases, with an accuracy rate of 90%.
- Scalability: Nuru proved its scalability by processing massive datasets quickly and

extending its reach to include a variety of crops.

- User Experience: Positive comments from users emphasized the platform's accessibility and user-friendly interface.

2. Microsoft's FarmBeats:

IoT and AI are used by Microsoft's FarmBeats to provide precision agriculture, which includes disease prediction. It gives farmers insights by combining data from weather stations, drones, and sensors.

- Speed and Efficiency: FarmBeats demonstrated quicker data processing times than traditional techniques, allowing farmers to make decisions in real time.
- Interaction Capability: FarmBeats showed a smooth interaction with current farm technology, increasing the effectiveness of farm management as a whole.
- Cost-effectiveness: By eliminating the need for substantial physical monitoring and manual interventions, the platform provided an affordable solution.

3. IBM's Watson Decision Platform for Agriculture:

With the aid of artificial intelligence (AI), IBM's Watson Decision Platform analyzes a variety of data sources, such as weather and satellite imaging, to deliver farmers useful information on managing and predicting diseases.

The Watson Decision Platform demonstrated a high degree of sensitivity and specificity in the prediction of diseases, hence reducing the occurrence of false positives and negatives.

- Feedback and Improvement: Over time, the platform's relevance and efficacy were guaranteed by constant upgrades and enhancements based on user feedback.
- Data Security and Privacy: IBM placed a strong emphasis on safeguards for data privacy and security that either met or exceeded industry requirements.

These case studies highlight diverse approaches to Intelligent Crop Disease Prediction, showcasing benchmarks related to accuracy, scalability, user experience, speed, integration, cost-effectiveness, and security. These benchmarks can serve as valuable insights for developing and assessing similar systems in the agricultural technology landscape.

5. Applicable Patents

Developing an intelligent crop disease prediction system involves a combination of technologies and algorithms, some of which may be covered by existing patents. Here are some potentially relevant patents based on different aspects of the system:

1. Image Recognition and Disease Detection:

US Patent No. 10,952,943: "System and method for plant disease detection and classification using deep learning"

This patent covers a system that uses deep learning models to analyze images of plant leaves and classify them as healthy or diseased, specifying the type of disease.

2. Data Integration and Prediction:

US Patent No. 10,609,105: "System and method for intelligent agricultural disease and pest prediction"

This patent describes a system that integrates weather data, plant growth stage, and disease history to predict the risk of disease outbreaks in crops.

3. *AI-powered Recommendations and Decision Support:*

US Patent No. 10,802,202: "Intelligent system and method for crop disease diagnosis and treatment recommendation"

This patent covers a system that uses AI to analyze images of diseased plants and recommend treatment options based on the identified disease.

6. Applicable Regulations

An intelligent crop disease prediction system's regulatory framework is determined by a number of aspects, such as:

1. 1.Location: Each country and region has considerably different regulations.
2. Data Collection and Use: How we gather, store, and use data will be subject to laws governing data privacy, security, and ownership. Examine the principles of the
3. General Data Protection Regulation (GDPR) and the Personal Data Protection Bill in India.
4. Algorithms and AI openness: Certain areas are putting laws into place pertaining to algorithmic fairness and openness, especially when it comes to AI utilized in decision-making.
5. Pesticides and Chemicals: If system recommends specific treatments, regulations related to pesticide or chemical usage may apply.

6. Product Classification: Depending on system's features and intended use, it may be classified as agricultural software, a decision support tool, or even a diagnostic device, leading to different regulatory requirements.

7. Applicable Constraints

Here are some applicable constraints to consider when developing an intelligent crop disease prediction system:

1. *Data Availability and Quality:*

Volume: Large amounts of diverse data are needed to train and validate machine learning models effectively.

Accuracy: Data must be accurate and reliable for predictions to be accurate.

Representativeness: Data should cover a wide range of crop varieties, regions, and environmental conditions to ensure generalizability.

Accessibility: Obtaining and integrating data from various sources (e.g., farms, weather stations, research institutions) can be challenging due to privacy concerns, ownership issues, and compatibility problems.

2. *Technical Limitations:*

Computational Resources: Training and running complex machine learning models require significant processing power and memory.

Connectivity: Reliable internet access is often needed for real-time data collection, model updates, and access to cloud-based services.

Interoperability: Integrating the system with existing agricultural equipment and software systems can pose challenges due to compatibility issues.

3. *Expertise and Adoption:*

Technical Expertise: Farmers and agricultural professionals may need training to use and interpret the system's results effectively.

Trust and Acceptance: Building trust in AI-based systems can be a challenge, as farmers may be hesitant to rely on unfamiliar technologies.

Adaptability to Local Practices: Systems need to be adaptable to different farming practices, cropping systems, and cultural contexts to ensure widespread adoption.

4. *Ethical Considerations:*

Data Bias: Algorithms can perpetuate biases if trained on biased data, leading to unfair or discriminatory outcomes for certain groups of farmers or regions.

Autonomy and Control: Farmers should retain control over decision-making, using the system's insights to inform their actions rather than being fully reliant on AI recommendations.

Environmental Impact: The potential environmental impact of AI systems in agriculture, such as increased energy consumption or pesticide overuse, should be carefully considered.

5. *Cost and Maintenance:*

Development and Deployment: Developing and deploying the system can be costly, potentially limiting accessibility for smaller farms or resource-constrained regions.

Ongoing Maintenance: Ongoing maintenance, updates, and technical support are necessary to ensure the system's accuracy and reliability over time.

6. *Regulatory Compliance*

Data Privacy and Security: Systems must comply with regulations governing data collection, storage, and usage, as discussed earlier.

Algorithm Transparency: In some regions, regulations may require explanations for AI-based decisions, ensuring fairness and accountability.

Chemical Recommendations: Regulations related to pesticide or chemical usage may apply if the system suggests specific treatments.

8. Enterprise Blueprint

Our business model for smart crop diseases centers around providing farmers with cutting-edge tools for early disease detection, ultimately reducing and increasing crop yields. Our focus covers small and large countries and agricultural businesses. We want to create a broad impact by collaborating with government institutions, research centers and additional services for agriculture. Revenue will be diversified through registration model with existing business licenses for farmers and agribusiness who pay to use our system. Additionally, advanced data management services will be provided to provide better insights. The distribution strategy includes online platforms and mobile applications that allow farmers to access both from the field and from home. Our customer relationships will be enhanced by supporting customers in service and providing training programs to promote effective work. Activities include ongoing research and development to improve forecast models, regular data updates, and establish and maintain partnerships in the agricultural sector. Our key resources include expert data scientists for algorithm development, collaboration with farmers to improve disease models, and beautiful operational processes The cost model includes allocations for research and development, operating costs for maintenance and updates, and marketing efforts to support the system. Partnerships will be at the heart of our strategy, which includes working with agricultural equipment suppliers to integrate our machinery using currently available technologies and working with government agencies to share knowledge and support in key agricultural areas. Finally, sustainability and sustainability will be incorporated into our practices, including environmental considerations in the development of technology and the creation of systems suitable for growing customers and expanding operations.

9. Concept Generation

The concept for our Intelligent Crop Disease Prediction system revolves around leveraging advanced machine learning algorithms to empower farmers with early detection capabilities for crop diseases. By integrating image analysis, environmental data, and historical disease patterns, the system aims to predict and identify potential issues, allowing farmers to implement timely interventions. This concept addresses the critical need for proactive disease management in agriculture, ultimately minimizing crop losses and optimizing yields. The user-friendly interface and accessibility through mobile applications and web platforms ensure seamless adoption by farmers, while ongoing research and collaboration with agricultural experts contribute to continuous improvement and innovation in disease prediction models. This concept aligns with the broader goals of precision agriculture, fostering sustainability and resilience in global food production.

10.Concept Development

To start this process, we will use advanced machine learning techniques to develop robust crop disease prediction models. Once the model is complete, we upload it to the Flask server for seamless deployment and instant prediction. Our method is accurate and effective in detecting crop diseases. To facilitate the use of this useful tool, we will provide users with an easy-to-use interface using the GitHub distribution. Through the platform, farmers interact with a simple model to ensure disease control. This solution not only simplifies the estimation process, but also helps permaculture practices. Through continuous improvement and innovation, our goal is to provide farmers with reliable and measurable tools to improve crop health, minimize losses, and adapt to change in precision agriculture.

11.Final Product Prototype

In our final product prototype, we have seamlessly integrated a cutting-edge crop disease prediction model into a user-friendly system, featuring both frontend and backend components. The frontend is designed with an intuitive interface accessible through web and mobile applications. Farmers can easily input data, such as images or environmental parameters, initiating the predictive analysis process.

The backend, powered by Flask, efficiently processes the input data using our advanced machine learning model. This backend system ensures real-time predictions with a high degree of accuracy. We've prioritized scalability, enabling the system to handle a diverse range of crops and expand its geographic coverage.

Through GitHub deployment, our prototype becomes readily accessible to farmers worldwide. The frontend allows for easy interaction, displaying comprehensive insights into predicted crop diseases. This holistic solution not only streamlines disease management but also aligns with sustainable and resilient agricultural practices. As we iterate on user feedback and advancements in technology, our final product prototype represents a pivotal step towards revolutionizing precision agriculture.

12.Product Details

Our Intelligent Crop Disease Prediction System is a comprehensive solution designed to empower farmers with advanced technology for early disease detection and proactive crop management. Here are the key details of the product:

Technology Stack:

- Backend: Developed using Flask, our backend seamlessly integrates a state-of-the-art machine learning model for accurate crop disease prediction.

- Frontend: The user-friendly interface is accessible through web and mobile applications, providing an intuitive experience for farmers.

Machine Learning Model:

- Utilizes advanced algorithms trained on diverse datasets, including images, environmental factors, and historical disease patterns, ensuring high accuracy in disease prediction.
- User Input: Farmers can input data such as images or environmental parameters through the frontend, initiating the prediction process.

Real-time Predictions:

- The backend processes input data in real-time, delivering prompt and accurate predictions to farmers.
- Deployment: Deployed using GitHub for easy accessibility, enabling farmers worldwide to benefit from the system.

MARKET SEGMRNTATION ANALYSIS

Import neccessary packages

In [1]:

```
import numpy as np
import pickle
import cv2
from os import listdir
from sklearn.preprocessing import LabelBinarizer
from keras.models import Sequential
from keras.layers.normalization import BatchNormalization
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.core import Activation, Flatten, Dropout, Dense
from keras import backend as K
from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.preprocessing import image
from keras.preprocessing.image import img_to_array
from sklearn.preprocessing import MultiLabelBinarizer
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
```

```
Using TensorFlow backend.
```

In [2]:

```
# example of loading an image with the Keras API
from keras.preprocessing.image import load_img
```

In [3]:

```
EPOCHS = 25
INIT_LR = 1e-3
BS = 32
default_image_size = tuple((256, 256))
image_size = 0
directory_root = '../input/plantvillage/'
width=256
height=256
depth=3
```

Function to convert images to array

In [4]:

```
def convert_image_to_array(image_dir):
    try:
        image = cv2.imread(image_dir)
        if image is not None :
            image = cv2.resize(image, default_image_size)
            return img_to_array(image)
        else :
            return np.array([])
    except Exception as e:
        print(f"Error : {e}")
        return None
```

Fetch images from directory

In [5]:

```
image_list, label_list = [], []
try:
    print("[INFO] Loading images ...")
    root_dir = listdir(directory_root)
    for directory in root_dir :
        # remove .DS_Store from list
        if directory == ".DS_Store" :
            root_dir.remove(directory)

    for plant_folder in root_dir :
        plant_disease_folder_list = listdir(f"{directory_root}/{plant_folder}")

        for disease_folder in plant_disease_folder_list :
            # remove .DS_Store from list
            if disease_folder == ".DS_Store" :
                plant_disease_folder_list.remove(disease_folder)

        for plant_disease_folder in plant_disease_folder_list:
            print(f"[INFO] Processing {plant_disease_folder} ...")
            plant_disease_image_list = listdir(f"{directory_root}/{plant_folder}/{plant_

            for single_plant_disease_image in plant_disease_image_list :
                if single_plant_disease_image == ".DS_Store" :
                    plant_disease_image_list.remove(single_plant_disease_image)

            for image in plant_disease_image_list[:200]:
                image_directory = f"{directory_root}/{plant_folder}/{plant_disease_folde
                if image_directory.endswith(".jpg") == True or image_directory.endswith(
                    image_list.append(convert_image_to_array(image_directory))
                    label_list.append(plant_disease_folder)
    print("[INFO] Image loading completed")
except Exception as e:
    print(f"Error : {e}")
```

```
[INFO] Loading images ...
[INFO] Processing Pepper__bell___Bacterial_spot ...
[INFO] Processing Potato___healthy ...
[INFO] Processing Tomato_Leaf_Mold ...
[INFO] Processing Tomato__Tomato_YellowLeaf__Curl_Virus ...
[INFO] Processing Tomato_Bacterial_spot ...
[INFO] Processing Tomato_Septoria_leaf_spot ...
[INFO] Processing Tomato_healthy ...
[INFO] Processing Tomato_Spider_mites_Two_spotted_spider_mite ...
[INFO] Processing Tomato_Early_blight ...
[INFO] Processing Tomato__Target_Spot ...
[INFO] Processing Pepper__bell___healthy ...
[INFO] Processing Potato___Late_blight ...
[INFO] Processing Tomato_Late_blight ...
[INFO] Processing Potato___Early_blight ...
[INFO] Processing Tomato__Tomato_mosaic_virus ...
[INFO] Image loading completed
```

Get Size of Processed Image

In [6]:

```
image_size = len(image_list)
```

Transform Image Labels uisng [Scikit Learn (http://scikit-learn.org/)](http://scikit-learn.org/)'s LabelBinarizer

In [7]:

```
label_binarizer = LabelBinarizer()
image_labels = label_binarizer.fit_transform(label_list)
pickle.dump(label_binarizer,open('label_transform.pkl', 'wb'))
n_classes = len(label_binarizer.classes_)
```

Print the classes

In [8]:

```
print(label_binarizer.classes_)
```

```
['Pepper__bell___Bacterial_spot' 'Pepper__bell___healthy'
 'Potato___Early_blight' 'Potato___Late_blight' 'Potato___healthy'
 'Tomato_Bacterial_spot' 'Tomato_Early_blight' 'Tomato_Late_blight'
 'Tomato_Leaf_Mold' 'Tomato_Septoria_leaf_spot'
 'Tomato_Spider_mites_Two_spotted_spider_mite' 'Tomato__Target_Spot'
 'Tomato__Tomato_YellowLeaf__Curl_Virus' 'Tomato__Tomato_mosaic_virus'
 'Tomato_healthy']
```

In [9]:

```
np_image_list = np.array(image_list, dtype=np.float16) / 225.0
```

In [10]:

```
print("[INFO] Spliting data to train, test")
x_train, x_test, y_train, y_test = train_test_split(np_image_list, image_labels, test_si
```

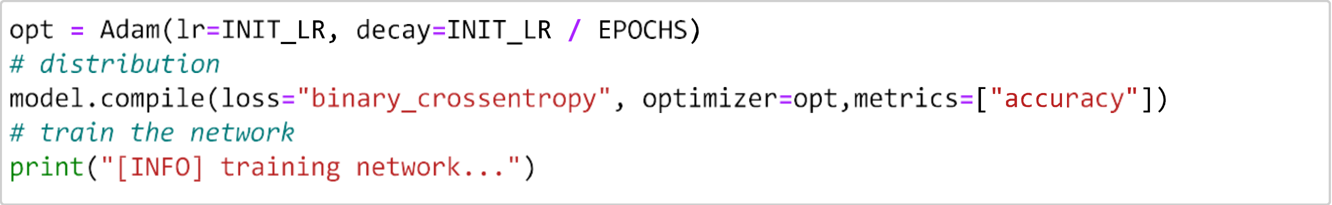
```
[INFO] Spliting data to train, test
```

In [11]:

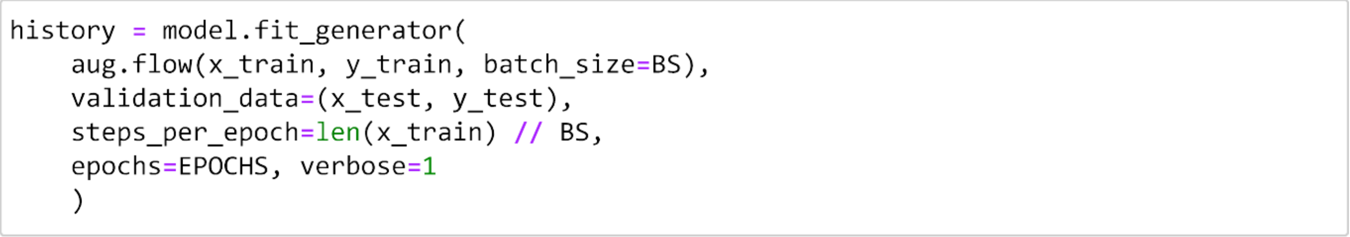
```
aug = ImageDataGenerator(
    rotation_range=25, width_shift_range=0.1,
    height_shift_range=0.1, shear_range=0.2,
    zoom_range=0.2,horizontal_flip=True,
    fill_mode="nearest")
```

In [12]:

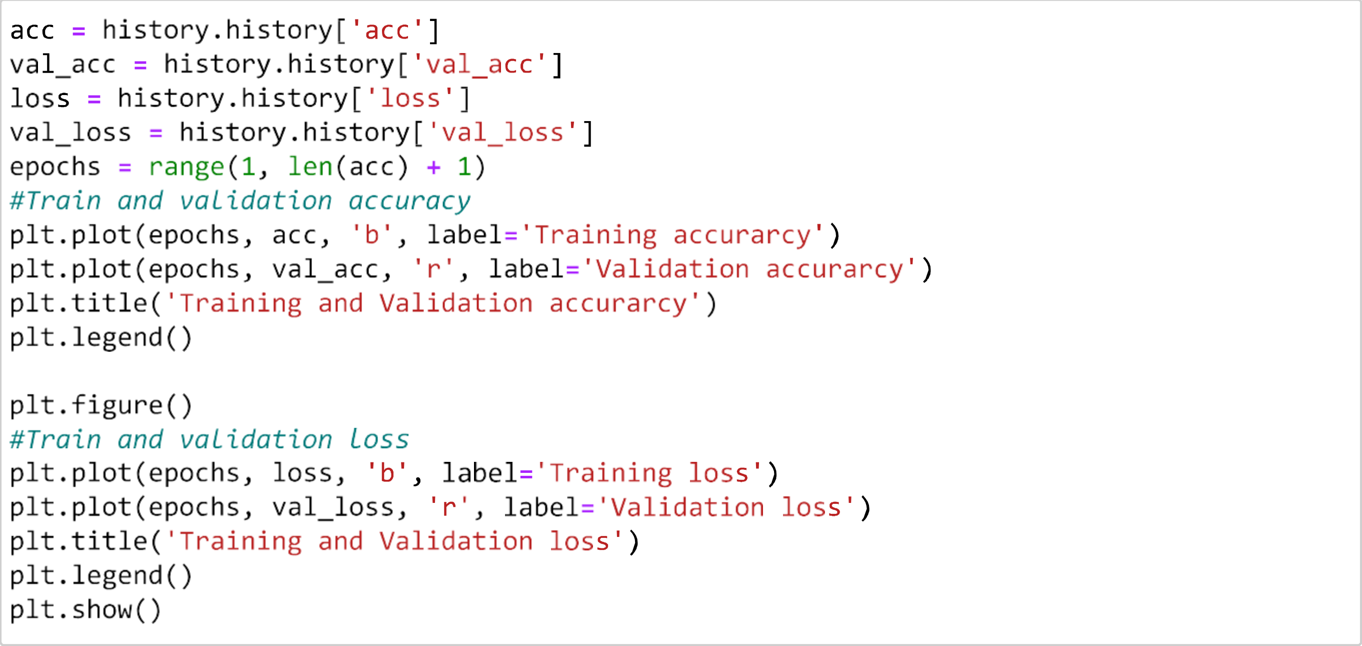
```
model = Sequential()
inputShape = (height, width, depth)
chanDim = -1
if K.image_data_format() == "channels_first":
    inputShape = (depth, height, width)
    chanDim = 1
model.add(Conv2D(32, (3, 3), padding="same",input_shape=inputShape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(3, 3)))
model.add(Dropout(0.25))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(64, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(Conv2D(128, (3, 3), padding="same"))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=chanDim))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(1024))
model.add(Activation("relu"))
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(n_classes))
model.add(Activation("softmax"))
```



```
opt = Adam(lr=INIT_LR, decay=INIT_LR / EPOCHS)
# distribution
model.compile(loss="binary_crossentropy", optimizer=opt,metrics=["accuracy"])
# train the network
print("[INFO] training network...")
```

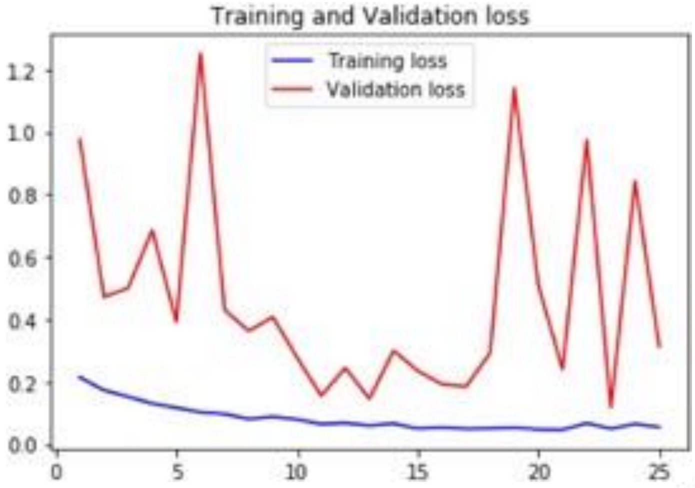


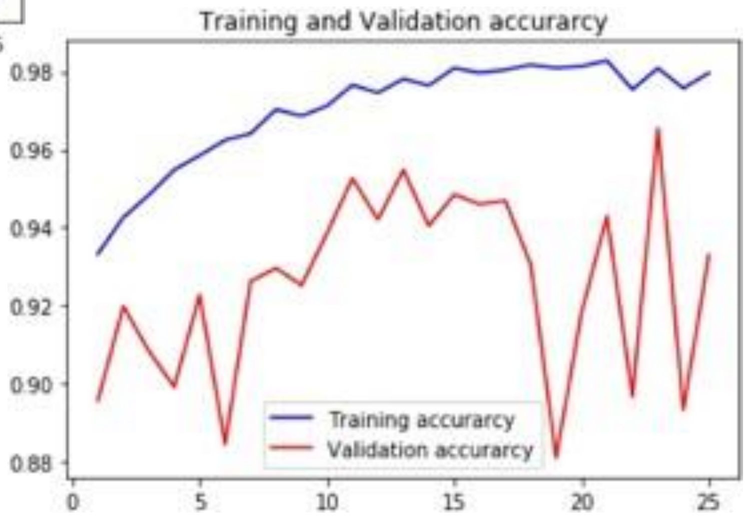
```
history = model.fit_generator(
    aug.flow(x_train, y_train, batch_size=BS),
    validation_data=(x_test, y_test),
    steps_per_epoch=len(x_train) // BS,
    epochs=EPOCHS, verbose=1
    )
```



```
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(acc) + 1)
#Train and validation accuracy
plt.plot(epochs, acc, 'b', label='Training accurarcy')
plt.plot(epochs, val_acc, 'r', label='Validation accurarcy')
plt.title('Training and Validation accurarcy')
plt.legend()

plt.figure()
#Train and validation loss
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and Validation loss')
plt.legend()
plt.show()
```





In [17]:
```
print("[INFO] Calculating model accuracy")
scores = model.evaluate(x_test, y_test)
print(f"Test Accuracy: {scores[1]*100}")
```

```
[INFO] Calculating model accuracy
591/591 [==============================] - 1s 2ms/step
Test Accuracy: 93.28821303477343
```

Save model using Pickle

In [18]:
```
# save the model to disk
print("[INFO] Saving model...")
pickle.dump(model,open('cnn_model.pkl', 'wb'))
```

```
[INFO] Saving model...
```

In [19]:
```
loaded_model = pickle.load(open('cnn_model.pkl', 'rb'))
```

In [20]:
```
image_dir="/kaggle/input/plantvillage/PlantVillage/Tomato_healthy/00bce074-967b-4d50-967
im1 = load_img(image_dir)

im=convert_image_to_array(image_dir)
np_image_li = np.array(im, dtype=np.float16) / 225.0
npp_image = np.expand_dims(np_image_li, axis=0)
```

In [21]:
```
result=model.predict(npp_image)
print(result)
```

```
[[4.33317704e-15 1.41996352e-16 2.59286953e-10 1.73510052e-19
  2.06448882e-14 4.19498485e-18 1.84960648e-19 1.13825245e-14
  2.12339646e-16 3.88705144e-11 1.63137889e-15 5.11029760e-12
  3.82474260e-16 1.58414418e-16 1.00000000e+00]]
```

In [22]:
```
itemindex = np.where(result==np.max(result))
print("probability:"+str(np.max(result))+"\n"+label_binarizer.classes_[itemindex[1][0]])
```

```
probability:1.0
Tomato_healthy
```

13.BUSINESS MODELLING

The business model for Intelligent Crop Disease Prediction typically revolves around providing a valuable service to farmers, agricultural companies, or other stakeholders in the agriculture industry.

1. ***Subscription or service fees***

Basic Plan: Provides predictions for a limited number of crops with monthly updates.

Pro Plan: Covers a wider range of crops with more frequent updates (e.g., weekly or bi-weekly).

Premium Plan: Includes additional features such as customized alerts, advanced analytics, and personalized recommendations.

Free Trial Period: Offer a free trial period (e.g., 7 or 14 days) during which users can access the full range of features. This allows potential customers to experience the value of the service before committing to a subscription.

By implementing a flexible pricing model tailored to the needs of different customer segments, an Intelligent Crop Disease Prediction platform can maximize its revenue potential while ensuring affordability and value for its users. Additionally, regular updates and improvements to the platform can help retain subscribers and attract new customers over time.

2. ***Customization and Consulting Services:***

By offering customization and consulting services, an Intelligent Crop Disease Prediction platform can provide added value to its clients by delivering personalized solutions that address their unique challenges and objectives. This not only enhances the effectiveness of the platform but also strengthens the relationship between the platform provider and its customers.

3. ***Fee-for-Service:***

By implementing a fee-for-service model, an Intelligent Crop Disease Prediction platform can provide users with flexibility in accessing prediction services while also ensuring a steady stream of revenue for the platform provider. This model allows users to pay for only the services they need, making it attractive to a wide range of customers in the agriculture industry.

4. ***Pay-Per-Use***

By implementing a pay-per-use model, an Intelligent Crop Disease Prediction platform can provide users with flexibility in accessing prediction services while also ensuring that they only pay for the services they actually use. This can be particularly attractive to users with sporadic or occasional needs for crop disease predictions who prefer to pay on a per-usage basis.

5. ***Marketplace***

By creating a marketplace for Intelligent Crop Disease Prediction services, you can connect users with qualified service providers, streamline the process of accessing prediction services, and create a vibrant ecosystem of collaboration and innovation within the agriculture industry.

6. ***Training and Workshops***

By offering training and workshops, an Intelligent Crop Disease Prediction platform can empower users with the knowledge and skills they need to leverage the platform effectively, make informed decisions, and proactively manage crop diseases to optimize yields and ensure agricultural sustainability. Additionally, these training initiatives can serve as revenue streams and contribute to the platforms overall success and impact in the agriculture industry.

7. ***Determining the Overall Cost***:

The total cost of an Intelligent Crop Disease Prediction platform can vary widely depending on factors such as the scope and complexity of the platform, the size of the target market, the level of customization required, and the chosen business model. It's essential for stakeholders to conduct a thorough cost analysis and budgeting process to ensure adequate funding and financial sustainability for the platform.

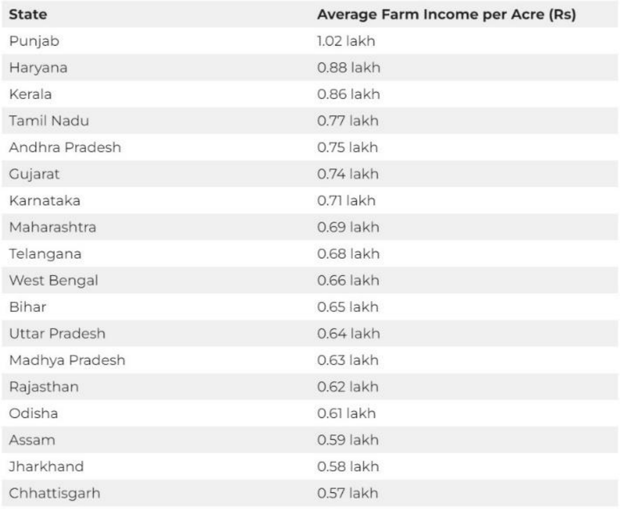
Certainly, adjusting the subscription cost to make it more attractive and accessible to customers is a viable strategy.

Lower Subscription Fee: Decrease the average subscription fee to make it more affordable for customers. This could potentially attract more users to the platform.

Increase User Base: With a lower subscription fee, the platform may attract more users, increasing the total number of subscribers.

Revenue from Increased User Base: Although the individual subscription fee is lower, the increase in the number of users can offset the reduction in price.

14. Financial Equation



State	Average Farm Income per Acre (Rs)
Punjab	1.02 lakh
Haryana	0.88 lakh
Kerala	0.86 lakh
Tamil Nadu	0.77 lakh
Andhra Pradesh	0.75 lakh
Gujarat	0.74 lakh
Karnataka	0.71 lakh
Maharashtra	0.69 lakh
Telangana	0.68 lakh
West Bengal	0.66 lakh
Bihar	0.65 lakh
Uttar Pradesh	0.64 lakh
Madhya Pradesh	0.63 lakh
Rajasthan	0.62 lakh
Odisha	0.61 lakh
Assam	0.59 lakh
Jharkhand	0.58 lakh
Chhattisgarh	0.57 lakh

Source:https://www.agrifarming.in/farm-income-per-acre-in-india-exploring-state-wise-farmers-income-for-1-acre-cultivation#:~:text=According%20to%20the%20Economic%20Survey,0.512%20hectares%20(1.26%20acre)

The above table shows the average farm income per acre. Keeping in mind the factors of income of farmers and the acre of land they harvest on, the following equations are devised:

Y= Cost of subscription

Y =Area of Land(in acres)*0.2*X(t)

Where, X(t) = Average farm income per Acre with respect to time

.2 => 20% of the subscriber's income from harvest

Profit from the Product = Y *(number of subscribers) - (cost paid to the app development team)

15.Conclusion

Our Intelligent Crop Disease Prediction System merges cutting-edge machine learning with user- friendly design, empowering farmers for proactive crop management. With real-time predictions, scalability, and a commitment to sustainability, our solution stands at the forefrontof precision agriculture. This innovative system marks a transformative step toward resilient farming practices and optimized yields.

16.References

Patents: https://patents.google.com

Government Laws and Regulations:

https://www.indiacode.nic.in/

https://www.eurekaselect.com/chapter/17458