# Plant Disease Detection Using CNN IN Plants

*-By Puppala Saiteja*

**Step-1 : Prototype Selection**

*Abstract*

This application aims to revolutionize plant disease management by leveraging the power of computer vision and deep learning. By utilizing Convolutional Neural Networks (CNNs), the app can accurately detect and classify various plant diseases based on images captured by farmers. This early detection enables timely intervention, preventing significant crop losses and improving overall agricultural productivity.

This plant disease detection app aims to empower farmers by providing a user-friendly tool for accurate and timely identification of plant diseases. By leveraging the power of Convolutional Neural Networks (CNNs), the app can detect diseases at an early stage, enabling farmers to take prompt action and prevent significant crop losses. This, in turn, contributes to improved agricultural efficiency, reduced reliance on chemical inputs, and ultimately, a more sustainable and productive agricultural system.

1. **Problem statement:**

Accurate and timely detection of plant diseases is a critical challenge in agriculture. Traditional methods of disease identification often rely on expert knowledge and visual inspection, which can be time-consuming, labor-intensive, and prone to errors. The increasing availability of Smartphone’s and digital imaging technologies has opened up new opportunities for developing automated disease detection systems.

**Leveraging Farmer-Provided Images**

By empowering farmers to capture images of diseased plants and upload them to a centralized platform, a wealth of data can be collected for analysis. This approach offers several advantages

* **Early detection**: Farmers can identify potential issues at early stages, allowing for timely interventions.
* **Data accessibility**: Large-scale datasets can be generated to train robust machine learning models.
* **Geographical coverage**: Data can be collected from diverse regions, capturing awide range of disease variations.

1. **CUSTOMER/MARKET/BUSINESS REQUIREMENTS**

Farmers are the primary customers for a disease detection system based on image analysis. Their needs can be categorized as follows:

* 1. Image Capture for Plant Disease Detection

* + 1. Crop focus: Ensure the plant or affected part is the primary subject.
    2. Multiple images: Capture multiple images of the same plant from different angles.
    3. Image metadata: Include relevant information such as crop type, location, and date.
    4. Image format: Use a suitable image format (e.g., JPEG, PNG) that maintains image quality.
  1. Disease Selection in Farms

Disease selection within a farming context presents a unique set of challenges. Farmers often encounter a wide variety of crops and diseases, each with distinct symptoms and impacts.

* + 1. Crop Diversity: The system must accommodate a broad range of crops, from major staples to niche or regional varieties.
    2. Disease Specificity: Accurate identification requires a comprehensive database of diseases affecting different crops.
    3. Symptom Variation: Diseases can manifest in various ways, making visual identification challenging.
    4. Disease Stages: The system should be able to detect diseases at different stages of development.
    5. User Interface: A user-friendly interface is essential for farmers to easily select the appropriate crop and disease.
  1. Potential Approaches:
     1. Hierarchical Classification: Organizing diseases into categories based on crop type, disease family, or symptoms can simplify the selection process.
     2. Image-Based Selection: Allowing users to select a disease based on visual similarity to reference images.
     3. Symptom-Based Selection: Providing a list of symptoms for users to choose from.
     4. Expert System: Incorporating expert knowledge to guide users in selecting the most likely disease.
  2. Disease Display: Providing Clear and Actionable Information

Effective disease display is crucial for empowering farmers to take appropriate actions. The information presented should be clear, concise, and easily understandable.

* + 1. Disease Name: Clearly identify the detected disease.
    2. Disease Image: Visual representation of the disease for comparison.
    3. Disease Symptoms: Detailed description of the disease symptoms to aid in verification.
    4. Disease Severity: Indication of the severity of the disease to prioritize treatment.
    5. Treatment Recommendations: Suggested control measures or treatments.
    6. Preventive Measures: Information on preventing the spread of the disease.
    7. Image Comparison: Display the uploaded image side-by-side with a reference image of the detected disease.
    8. Symptom Highlighting: Indicate specific symptoms identified in the uploaded image.
    9. Disease Progression: Visualize the potential progression of the disease if left untreated.

1. **Target Specifications and Characterizations**

Understanding the target customer is essential for developing a successful disease detection system. This involves identifying specific customer segments, characterizing their needs, and defining clear specifications for the system.

* 1. **Customer Segmentation**
     1. Small-scale farmers: These farmers typically have limited resources and rely heavily on traditional farming practices
     2. Large-scale commercial farmers: These farmers operate on a larger scale and may have access to more advanced technologies.
     3. Agricultural cooperatives: These organizations represent groups of farmers and can facilitate system adoption.
     4. Government agricultural extension services: These agencies can use the system for disease surveillance and early warning systems.
  2. **Customer Characterizations**
     1. Technical proficiency: Varies widely across different customer segments.
     2. Smartphone ownership: High Smartphone penetration among younger farmers.
     3. Internet access: Availability of reliable internet connectivity varies by region.
     4. Agricultural knowledge: Level of understanding of plant diseases and crop management practices.
     5. Decision-making processes: How farmers make decisions regarding crop management and disease control.

By carefully considering these customer specifications and characterizations, we can develop a disease detection system that effectively meets the needs of different user groups and drives adoption.

1. **EXTERNAL SEARCH**

While specific news on disease prediction using farmer-provided images might be limited, there's a surge in AI applications across the agriculture sector. Here are some recent trends:

* 1. AI in Agriculture: A Growing Trend
     1. Drones and Image Analysis: Companies and research institutions are increasingly using drones equipped with high-resolution cameras to capture agricultural data. This data is then analyzed using AI to monitor crop health and identify potential problems, including disease outbreaks. Identify crop health issues, including disease detection.
     2. Smartphone Apps for Farmers: Several apps are being developed to assist farmers in various tasks, including disease identification through image analysis. For example, some apps allow farmers to upload pictures of affected crops and receive potential diagnoses.
     3. AI-Powered Crop Monitoring: Companies are developing AI-driven platforms that use satellite imagery and machine learning.

1. **BENCH MARKING ALTERNATE PRODUCT OR PLATFORM**

There are existing products and platforms that leverage image-based disease prediction**.** While the specific focus on farmer-provided images might be relatively new, the underlying technology of image analysis for disease detection has been in development for several years.

* 1. **EXISTING PLATFORM AND PRODUCTS**:
     1. Agricultural Technology Startups: Many startups have developed mobile applications that allow farmers to capture images of diseased plants and receive potential diagnoses. Some of these platforms also offer additional services like expert consultations and treatment recommendations.
     2. Research Institutions and Universities: Several academic institutions have developed prototypes and proof-of-concept systems for disease detection using image analysis. While these might not be commercially available products, they can serve as benchmarks for your research.
     3. Large Agricultural Companies: Some established agricultural companies have integrated image-based disease detection into their digital farming platforms. These platforms often offer a suite of tools for farmers, including disease diagnosis, weather forecasting, and crop management.

1. **APPLICABLE REGULATIONS**

The development and deployment of a disease prediction system using farmer-provided images is subject to a complex interplay of regulations. These regulations are primarily concerned with data privacy, data security, and product safety.

* 1. **Data Privacy and Security:**

6.1.1 GDPR (General Data Protection Regulation): If you operate within the EU or handle data of EU residents, GDPR will apply. It mandates stringent data protection measures, including obtaining explicit consent, data minimization, and data breach notification.

6.1.2 CCPA (California Consumer Privacy Act): Similar to GDPR, but focused on California residents.

6.1.3 HIPAA (Health Insurance Portability and Accountability Act): While not directly applicable to agriculture, it provides a framework for handling sensitive health information, which can be relevant in certain cases.

6.1.4 Local Data Protection Laws: Many countries have their own data protection laws, such as India's IT Act.

* 1. **Intellectual Property:**

6.2.1 Copyright and Trademark: Protect your software, branding, and other intellectual assets.

6.2.2 Patent Law: If your system involves novel inventions, consider patent protection.

* 1. **Best Practices:**

6.3.1 Conduct a Thorough Regulatory Assessment: Identify all applicable regulations and assess their impact on your system.

6.3.2 Implement Strong Data Protection Measures: Use encryption, access controls, and regular security audits.

6.3.3 Obtain Legal Counsel: Seek expert advice to ensure compliance with complex legal requirements.

* + 1. Stay Updated: Keep abreast of changes in regulations and industry standards.

1. **APPLICABLE LIMITATIONS**
   1. **Data Quality Issues**
      1. Image Quality: Variations in lighting, focus, and angle can significantly impact the accuracy of disease detection.
      2. Data Consistency: Ensuring consistent image formats, resolutions, and metadata can be challenging.
      3. Data Quantity: Building a robust model requires a large and diverse dataset, which might be difficult to obtain.
   2. **Model Limitations**
      1. Disease Complexity: Some diseases exhibit similar symptoms, making accurate differentiation difficult.
      2. Disease Stages: Early-stage diseases might be challenging to detect due to subtle symptoms.
      3. Environmental Factors: Weather conditions, soil types, and other environmental factors can influence disease symptoms and impact model accuracy.
2. **BUSINESS OPPORTUNITIES**

Developing a disease prediction model using farmer-provided images presents several business opportunities:

* 1. **Direct Revenue Generation:**
     1. Software as a Service (SaaS): Offer a subscription-based model for farmers to access the disease prediction service.
     2. Pay-per-use: Charge farmers a fee for each image analysis and diagnosis.
     3. Enterprise Solutions: Partner with agricultural cooperatives or large-scale farming operations for enterprise-level solutions.
     4. Data Licensing: License the image data and derived insights to research institutions or companies for further development.
  2. **Indirect Revenue Generation:**
     1. Hardware Sales: Bundle the software with imaging devices (smartphones, tablets) or sensors.
     2. Value-Added Services: Offer additional services like crop insurance, agricultural inputs, or financial services.
     3. Data Monetization: Utilize anonymized and aggregated data for market research, insights, and product development.
     4. Partnerships: Collaborate with agricultural companies, input suppliers, or insurance providers for cross-selling opportunities.
  3. **Potential Business Models:**
     1. Direct-to-Farmer: Sell the disease prediction service directly to farmers through online platforms or mobile apps.
     2. Platform-as-a-Service (PaaS): Provide a platform for other businesses to build applications on top of your disease detection technology.
     3. Data Analytics Service: Offer data-driven insights and recommendations to agricultural stakeholders.

1. **CONCEPT GENARATION**

Concept generation involves brainstorming innovative ideas to address the problem and create a unique value proposition. Here are some potential concepts for a disease prediction system using farmer-provided images:

* 1. **Concepts:**
     1. Early Warning System: Focus on detecting diseases at their early stages to minimize crop loss.
     2. Expert System Integration: Combine AI-powered diagnosis with expert knowledge for enhanced accuracy.
     3. Community-Driven Knowledge Base: Build a knowledge base of diseases and their management practices based on farmer contributions.
     4. Mobile-First Approach: Design a mobile app with a user-friendly interface for easy image capture and diagnosis.
  2. **Value-Added Concepts:**
     1. Crop Insurance Integration: Partner with insurance companies to offer discounted premiums for users of the system.
     2. Financial Services: Provide microloans or credit facilities to farmers based on crop health assessments.
     3. Agricultural Input Recommendations: Suggest appropriate fertilizers, pesticides, or other inputs based on disease diagnosis.
     4. Market Linkage: Connect farmers with buyers or processors based on crop health and quality.
     5. Data Analytics Platform: Offer data-driven insights on disease prevalence, crop health trends, and regional patterns.
  3. **Innovative Concepts:**
     1. AI-Powered Crop Advisors: Provide personalized recommendations for crop management based on disease history, weather conditions, and soil data.
     2. Drone Integration: Use drones to capture high-resolution images for improved disease detection and crop monitoring.
     3. Virtual Reality Training: Offer immersive training modules for farmers on disease identification and management.
     4. Blockchain Technology: Ensure data security and transparency through blockchain-based data management.

1. **Concept Development:**

A mobile application that enables farmers to capture images of diseased plants, receives real-time diagnosis, and access relevant treatment recommendations.

* 1. **Features:**
     1. Image Capture: High-quality image capture with clear guidelines.
     2. Disease Identification: AI-powered model for accurate disease diagnosis.
     3. Treatment Recommendations: Expert-curated recommendations based on disease identification.
     4. Knowledge Base: Access to information on various diseases, symptoms, and prevention measures.
     5. Community Forum: Platform for farmers to share experiences and knowledge.
  2. **Value Proposition:**
     1. Timely Intervention: Early detection of diseases to minimize crop losses.
     2. Cost Reduction: Reduced reliance on chemical inputs and expert consultations.
     3. Improved Crop Yield: Enhanced crop health and productivity.
     4. Empowerment of Farmers: Increased knowledge and decision-making capabilities.
  3. **Market:**
     1. Small-scale and marginal farmers.
     2. Large-scale commercial farmers.
     3. Agricultural cooperatives and extension services.
  4. **Revenue Model:**
     1. Subscription-based access to the platform.
     2. Pay-per-use model for image analysis.
     3. Partnership with agricultural input providers for revenue sharing.
  5. **Technology Stack:**
     1. Mobile app development (iOS and Android).
     2. Cloud-based image processing and machine learning platform.
     3. AI algorithms for image analysis and disease classification.
     4. Secure data storage and management.

1. **FINAL PRODUCT PROTOTYPE:**

*Abstract*

This application aims to revolutionize plant disease management by leveraging the power of computer vision and deep learning. By utilizing Convolutional Neural Networks (CNNs), the app can accurately detect and classify various plant diseases based on images captured by farmers. This early detection enables timely intervention, preventing significant crop losses and improving overall agricultural productivity.

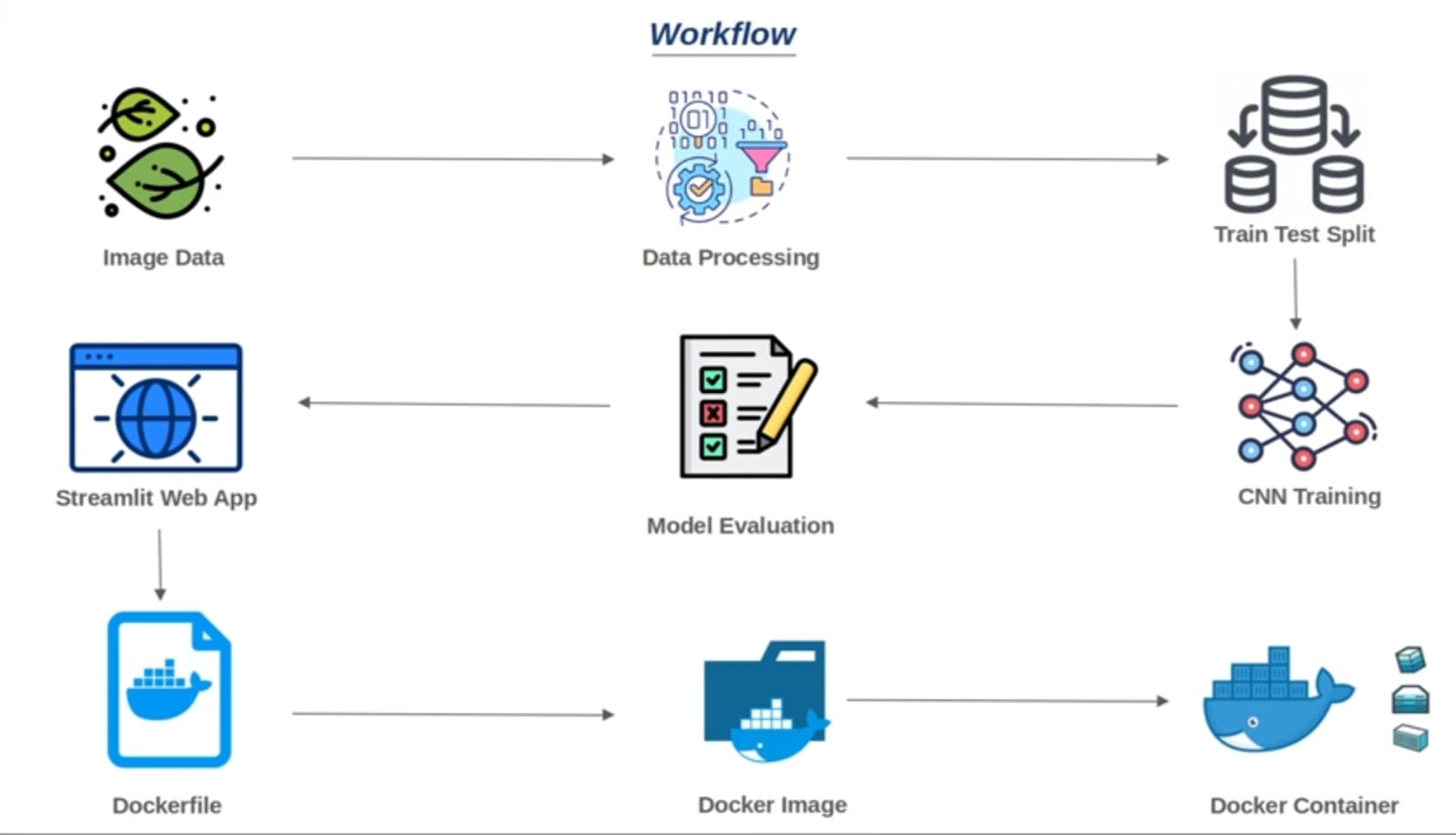
This plant disease detection app aims to empower farmers by providing a user-friendly tool for accurate and timely identification of plant diseases. By leveraging the power of Convolutional Neural Networks (CNNs), the app can detect diseases at an early stage, enabling farmers to take prompt action and prevent significant crop losses. This, in turn, contributes to improved agricultural efficiency, reduced reliance on chemical inputs, and ultimately, a more sustainable and productive agricultural system.

* 1. **Data Collection and Preprocessing**
     1. **Image Acquisition:** Collect a diverse dataset of plant images, including healthy and diseased plants from various species and stages of growth.
     2. **Data Annotation:** Label each image with the corresponding disease or healthy category.
     3. **Data Augmentation:** Apply techniques like image flipping, rotation, cropping, and color variations to increase dataset size and improve model robustness.
     4. **Preprocessing:** Standardize images by resizing, normalizing, and converting to a suitable format (e.g., RGB).
  2. **Model Training**
     1. **Data Split:** Divide the dataset into training, validation, and testing sets.
     2. **Model Initialization:** Initialize the EfficientNet model with appropriate parameters.
     3. **Transfer Learning:** Consider using pre-trained weights from a similar domain (e.g., ImageNet) to accelerate training.
     4. **Training:**
* Iterate over the training dataset, feeding images and labels to the model.
* Calculate the loss between predicted and actual labels.
* Update model weights using back propagation.
* Use techniques like dropout and regularization to prevent overfitting.
  + 1. **Validation:** Evaluate the model's performance on the validation set to monitor progress and adjust hyper parameters.
    2. **Hyper parameter Tuning:** Experiment with different hyper parameters (e.g., learning rate, batch size) to optimize model performance.
  1. **Deployment**
     1. **Model Conversion:** Convert the trained model to a suitable format (e.g., TensorFlow Lite, Core ML) for deployment on mobile devices.
     2. **App Integration:** Integrate the model into the plant disease detection app.
     3. **Image Processing:** When a user uploads an image, preprocess it using the same techniques applied during training.
     4. **Model Inference:** Feed the preprocessed image to the deployed model for prediction.
     5. **Result Display:** Present the predicted disease label and any additional information (e.g., treatment recommendations) to the user.

**Data collected :** [PlantVillage Dataset (kaggle.com)](https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset)

**You can find the Data loading through API , Data preprocessing, Model Training in this link** : <https://drive.google.com/drive/folders/1_fKJ2Un-PBmmY0Pm92p0IINI63d34WOi?usp=sharing>

* 1. **Schematic Diagram**

****

* + 1. **Image Data:** The process starts with collecting and preprocessing images of healthy and diseased plants. These images serve as the input data for the model.
    2. **Data Processing:** The collected images undergo various preprocessing steps, such as resizing, normalization, and augmentation, to prepare them for model training.
    3. **Train Test Split:** The dataset is divided into training and testing sets. The training set is used to train the CNN model, while the testing set is used to evaluate its performance.
    4. CNN Training: The CNN model is trained on the training dataset. This involves feeding the images to the model, calculating the loss between the predicted and actual labels, and updating the model's parameters using backpropagation.
    5. **Model Evaluation:** The trained model is evaluated on the testing set to assess its accuracy and performance. Metrics like accuracy, precision, recall, and F1-score can be used to evaluate the model's effectiveness.
    6. **Streamlit Web App:** A Streamlit web app is developed to provide a user interface for uploading images and receiving predictions.
    7. **Dockerfile:** A Dockerfile is created to define the environment and dependencies required to run the application.
    8. **Docker Image:** The Dockerfile is used to build a Docker image, which contains all the necessary components to run the application.
    9. **Docker Container:** The Docker image is deployed as a Docker container, providing a portable and isolated environment for the application.

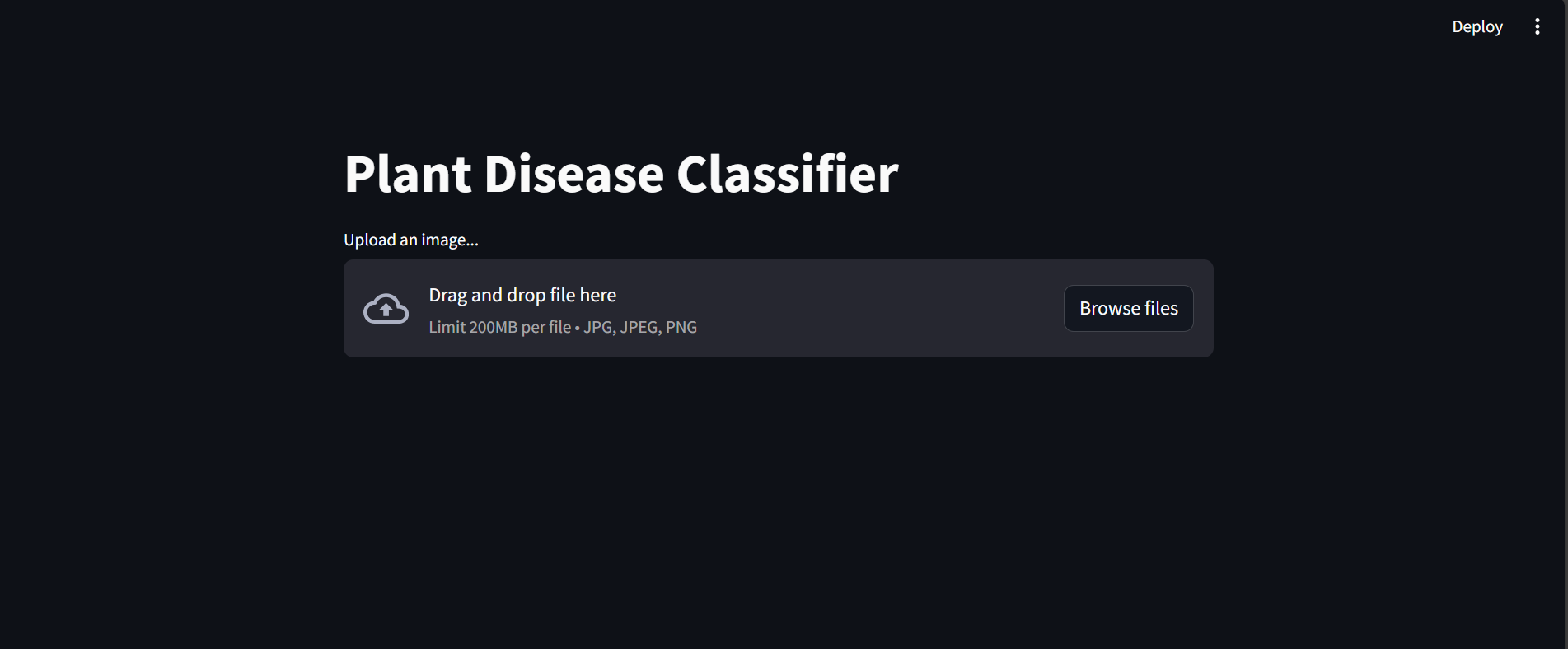
1. **PRODUCT DETAILS**
   1. **How does it work?**

The system operates on a basic principle of image recognition and classification. Farmers capture images of diseased plants using their smartphones. Then farmers use this app to recognize the disease of the plant by uploading their captured image.

By clicking on Analyze Button the model recognizes the structure of leaf to find which plant does the leaf belong to.

The model learns to identify patterns and features in the images that are indicative of specific diseases. When a new image of a plant is uploaded, the CNN analyzes the image and compares it to the patterns it has learned. Based on these comparisons, the model can predict whether the plant is healthy or diseased, and if it is diseased, it can identify the specific disease. This process allows for rapid and accurate diagnosis of plant diseases, enabling farmers to take timely action to prevent crop loss and improve yields.

Here is the frontend of the web app

****

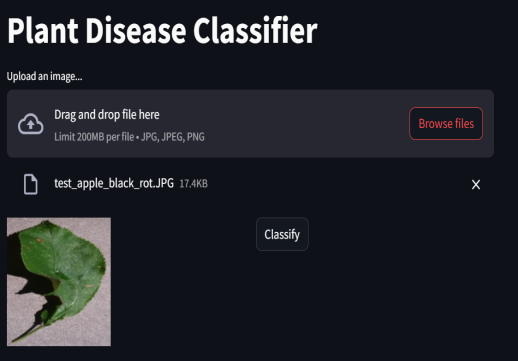
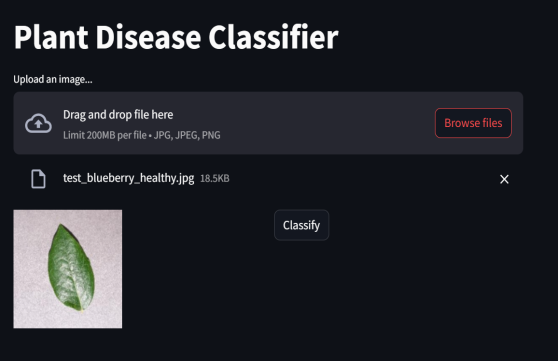
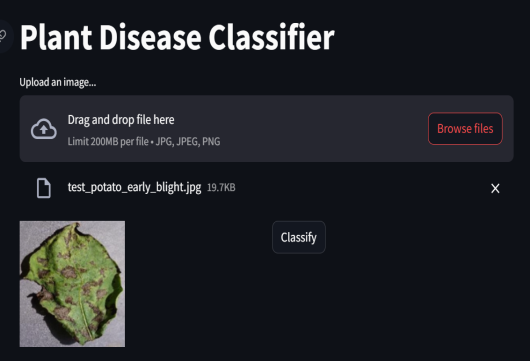
By clicking on browse file we can add image from our device.

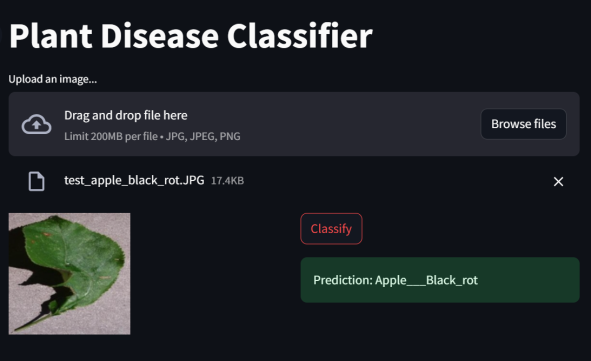
Lets test with some sample images



Lets upload these images and see the result.

After uploading the image we can find the classify button after clicking on the button the model predicts the disease of the plant.



Lets see the result after clicking on the classify button.

By absorbing the images we can say that the model is trained to detect the healthy plant leaf and diseased plant leaf.

* 1. DATA SOURSES
     1. **Farmer-Provided Images**:

Direct Collection: Encourage farmers to submit images of diseased plants.

Crowd sourcing: Utilize platforms to gather images from a wider user base.

Partnerships: Collaborate with agricultural organizations or cooperatives to collect data.

* + 1. **Research Institutions and Universities:**

Public Datasets: Many institutions share publicly available datasets for research purposes.

Collaborations**:** Partner with researchers to access their image collections.

* + 1. **Government Agricultural Departments:**

Historical Archives: Utilize existing image archives of plant diseases.

Current Data Collection**:** Collaborate with government agencies to collect new images.

* + 1. **Image Databases:**

Open-Source Platforms: Utilize platforms like Kaggle or ImageNet for publicly available datasets.

* 1. ALGORITHUM, FRAMEWORKS AND SOFTWARE MINIMUM REQUIREMENTS
     1. Algorithms:

**Convolutional Neural Networks (CNNs):** The most commonly used algorithm for image-based tasks, CNNs excel at extracting features from images. Architectures like ResNet, VGG, and Inception have shown promising results in plant disease detection.

**Transfer Learning**: Leveraging pre-trained models (e.g., ImageNet) can accelerate training and improve accuracy, especially with limited data.

**Object Detection Algorithms**: For detecting multiple diseases or lesions within an image, algorithms like YOLO or Faster R-CNN can be employed.

**Ensemble Methods**: Combining multiple models (e.g., Random Forest, Gradient Boosting) can enhance overall performance.

* + 1. Frameworks:

**TensorFlow and Keras:** Popular open-source platforms for building and training deep learning models. Offer a wide range of tools and flexibility.

**PyTorch:** Known for its ease of use and dynamic computational graph, PyTorch is suitable for rapid prototyping and research.

**OpenCV:** Primarily for computer vision tasks, OpenCV provides image processing functionalities and can be integrated with deep learning frameworks.

* + 1. Software and Tools:

**Python**: The predominant language for machine learning and data science, offering a rich ecosystem of libraries.

**Jupyter Notebook:** An interactive environment for data exploration, model development, and visualization.

**Cloud Platforms:** AWS, GCP, or Azure for scalable computing resources and machine learning services.

**Version Control**: Git for managing code and tracking changes.

* 1. TEAM REQUIRED
     1. Data Scientists/Machine Learning Engineers: Responsible for model development, training, and optimization.
     2. Image Processing Experts: Handle image preprocessing, feature extraction, and image analysis techniques.
     3. Software Engineers: Develop the mobile app, backend infrastructure, and user interface.
     4. Agricultural Scientists/Domain Experts: Provide expertise on plant diseases, symptoms, and treatment recommendations.
  2. WHAT DOES IT COST?

The cost of developing a disease detection system using farmer-provided images can vary significantly based on several factors including the complexity of the system, team size, technology stack, and geographic location. The following estimate is a rough approximation and should be used as a starting point.

* + 1. **Cost of Development and Maintenance:**

Estimating the cost of developing and maintaining a plant disease detection model using machine learning involves several factors:

**Data Acquisition and Preparation:** Cost of collecting, cleaning, and labeling plant image data.

**Model Development:** Costs associated with hiring data scientists, using cloud computing resources, and acquiring necessary software and hardware.

**Model Training:** Computational costs for training the machine learning model.

**Deployment and Maintenance:** Costs of hosting the model, updating it with new data, and ensuring its performance.

These costs can vary significantly depending on the complexity of the model, the scale of the project, and the resources used.

* + 1. **Revenue and Profit:**

Using the given equation, we can calculate the revenue and profit for different sales levels. For example, if the model is used by 1000 farmers per month, the revenue would be:

To calculate the profit, we need to subtract the development and maintenance costs from the revenue. Let's assume these costs are Rs. 100,000 per month.

* + 1. **Understanding the Revenue:**

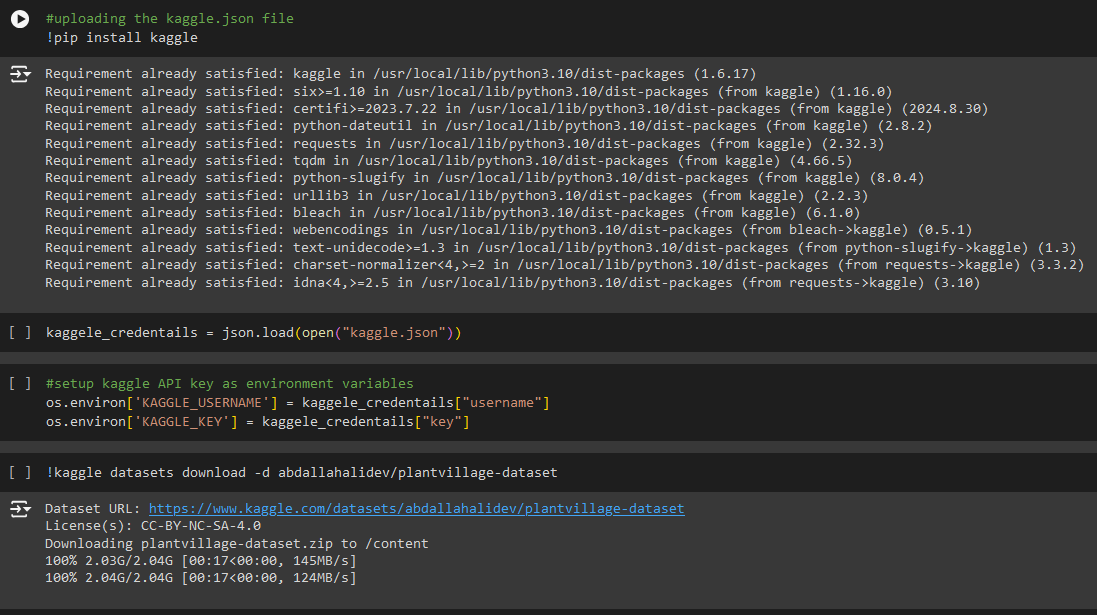
The equation you've provided, y = 500x - 2000, represents a simple linear relationship between revenue (y) and sales (x). The coefficient 500 represents the price per unit, and the constant -2000 represents the fixed costs (in this case, the monthly running costs).

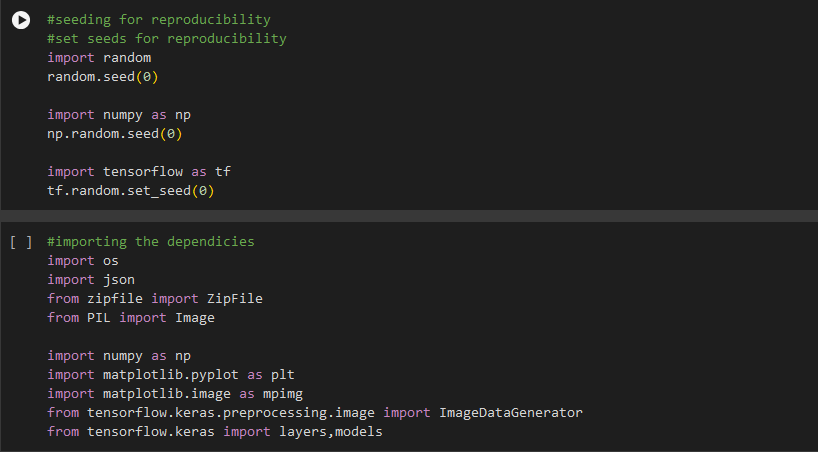
y = 500 \* 1000 - 2000 = Rs. 498,000

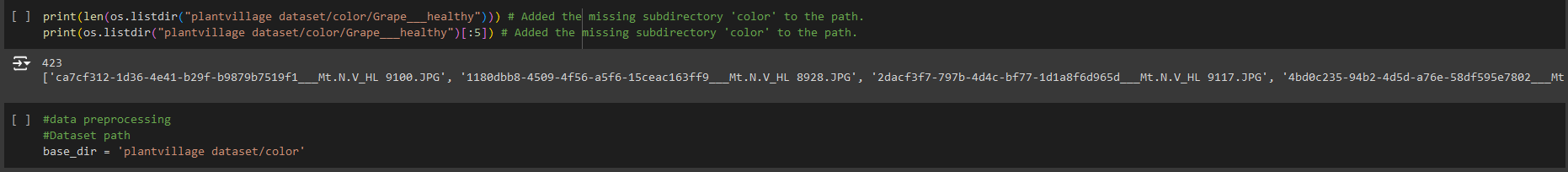
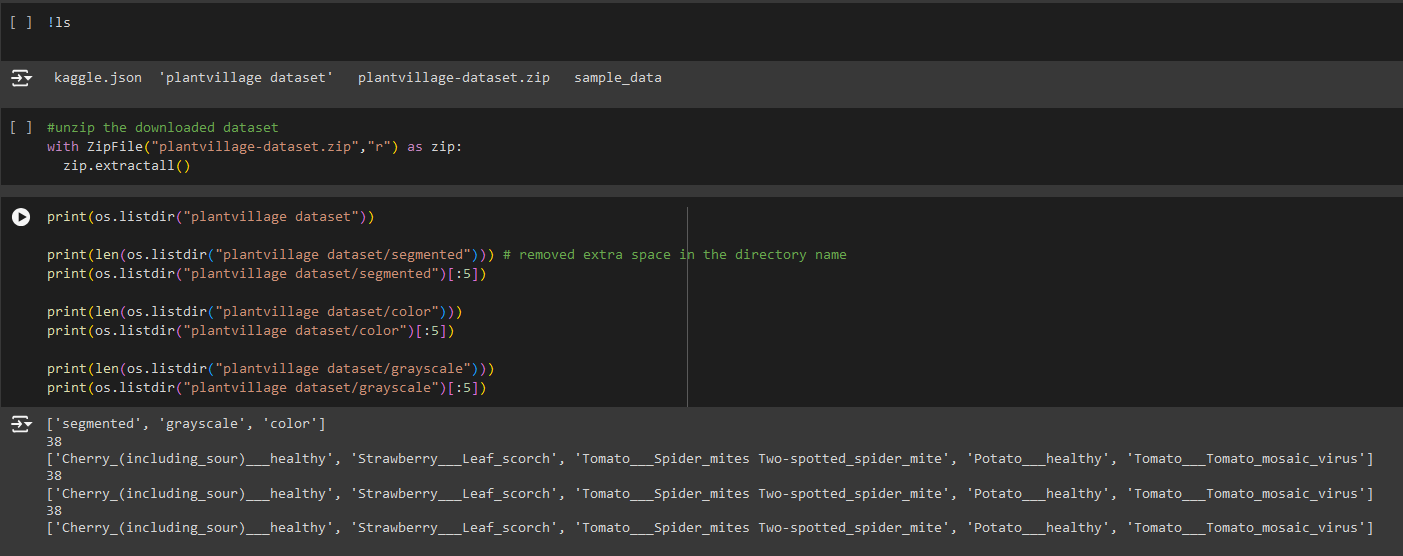
To calculate the profit, we need to subtract the development and maintenance costs from the revenue. Let's assume these costs are Rs. 100,000 per month.

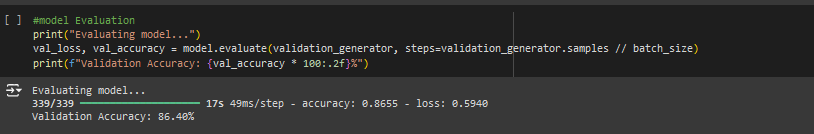
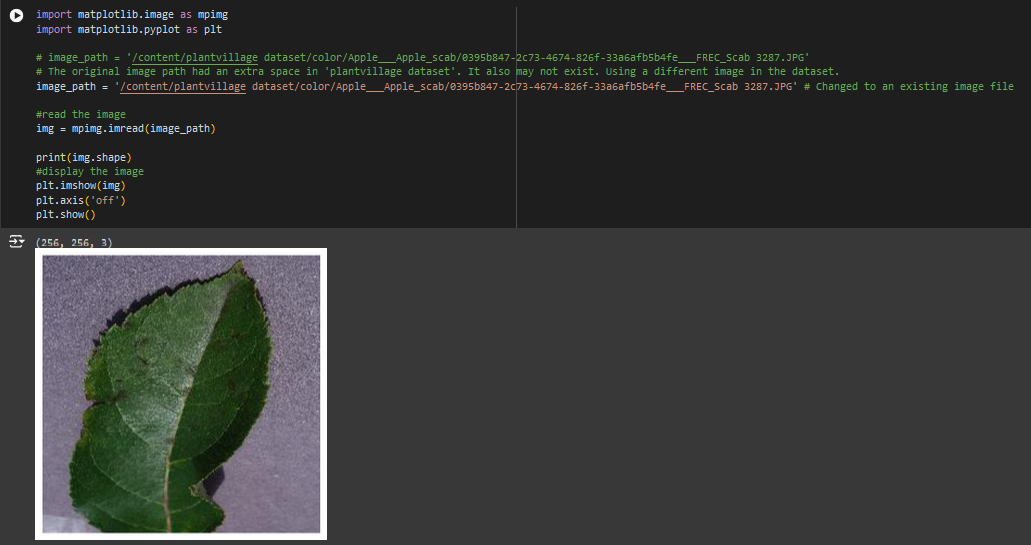
Profit = Revenue - Costs

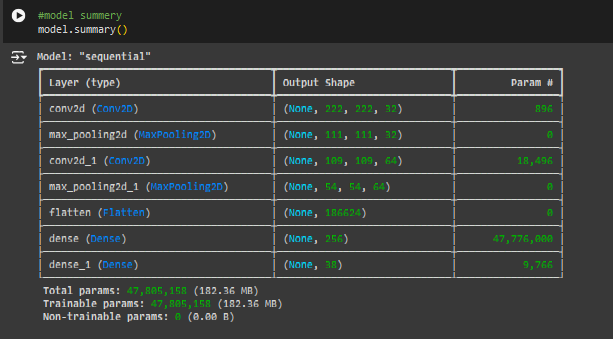
Profit = 498,000 - 100,000 = Rs. 398,000

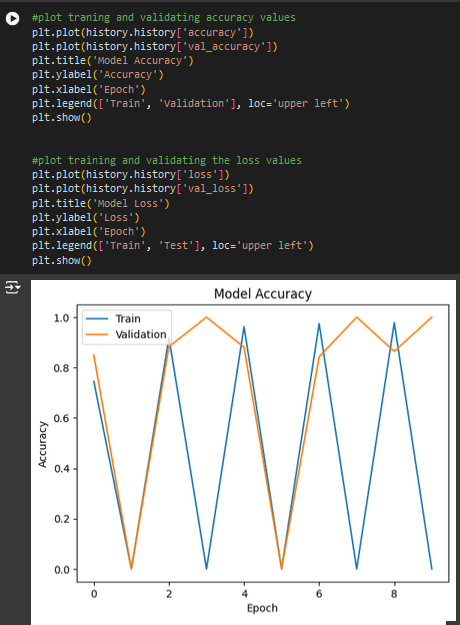
1. **Code Implementation/Validation**
   1. Sample EDA

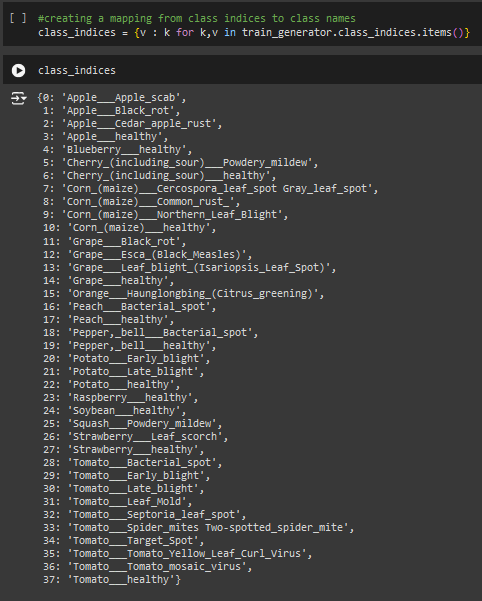
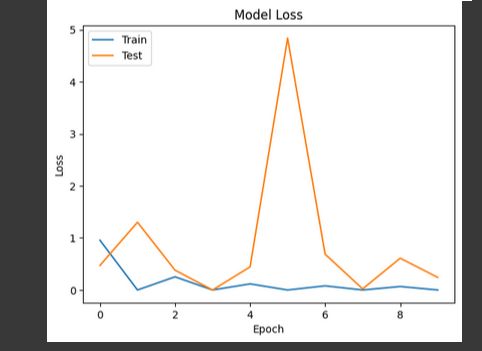












**Github link :**