DEEP LEARNING MODEL FOR DETECTING DISEASE IN TEA LEAVES

1.INTRODUCTION

1.1. PROJECT OVERVIEW

Tea plants are susceptible to various diseases that can negatively impact their growth and productivity. Early detection of these diseases is crucial for effective disease management and prevention.The proposed deep learning model will be trained on a large dataset of tea leaf images, containing both healthy leaves and leaves affected by various diseases. The model will learn to extract relevant features and patterns from the images, enabling it to classify tea leaves into different disease categories accurately.The development process will involve several key steps, including data collection and pre-processing, model architecture design, model training, and evaluation. A diverse dataset of tea leaf images will be collected, ensuring it represents different disease types, leaf stages, and environmental conditions. The collected dataset will then be pre-processed to enhance image quality, remove noise, and standardize the data format.For the model architecture, a deep convolutional neural network (CNN) will be utilized due to its ability to effectively capture spatial dependencies in images. The CNN will consist of multiple layers, such as convolutional layers, pooling layers, and fully connected layers, enabling the model to learn hierarchical representations of tea leaf images.To train the model, the pre-processed dataset will be divided into training and validation sets. The model will be trained using various optimization techniques, such as stochastic gradient descent (SGD) or Adam optimizer, with the objective of minimizing a chosen loss function (e.g., categorical cross-entropy). The model will iteratively adjust its parameters to optimize its performance on the training data.During training, techniques like data augmentation may be employed to increase the model's generalization ability by artificially expanding the dataset through techniques such as image rotation, flipping, and scaling. Regularization techniques like dropout may also be applied to prevent overfitting.After training, the model will be evaluated using a separate test dataset that was not used during training. Evaluation metrics such as accuracy, precision, recall, and F1 score will be used to assess the model's performance in disease detection.Once the model has achieved satisfactory performance, it can be deployed in practical scenarios, such as an application or a web-based interface. Users will be able to upload images of tea leaves, and the model will provide predictions on the presence and type of diseases present in the leaves.Overall, this project aims to leverage deep learning techniques to develop an accurate and efficient disease detection system for tea leaves, enabling early identification and timely intervention to prevent the spread of diseases and improve tea plant health and productivity.

1.2. PURPOSE

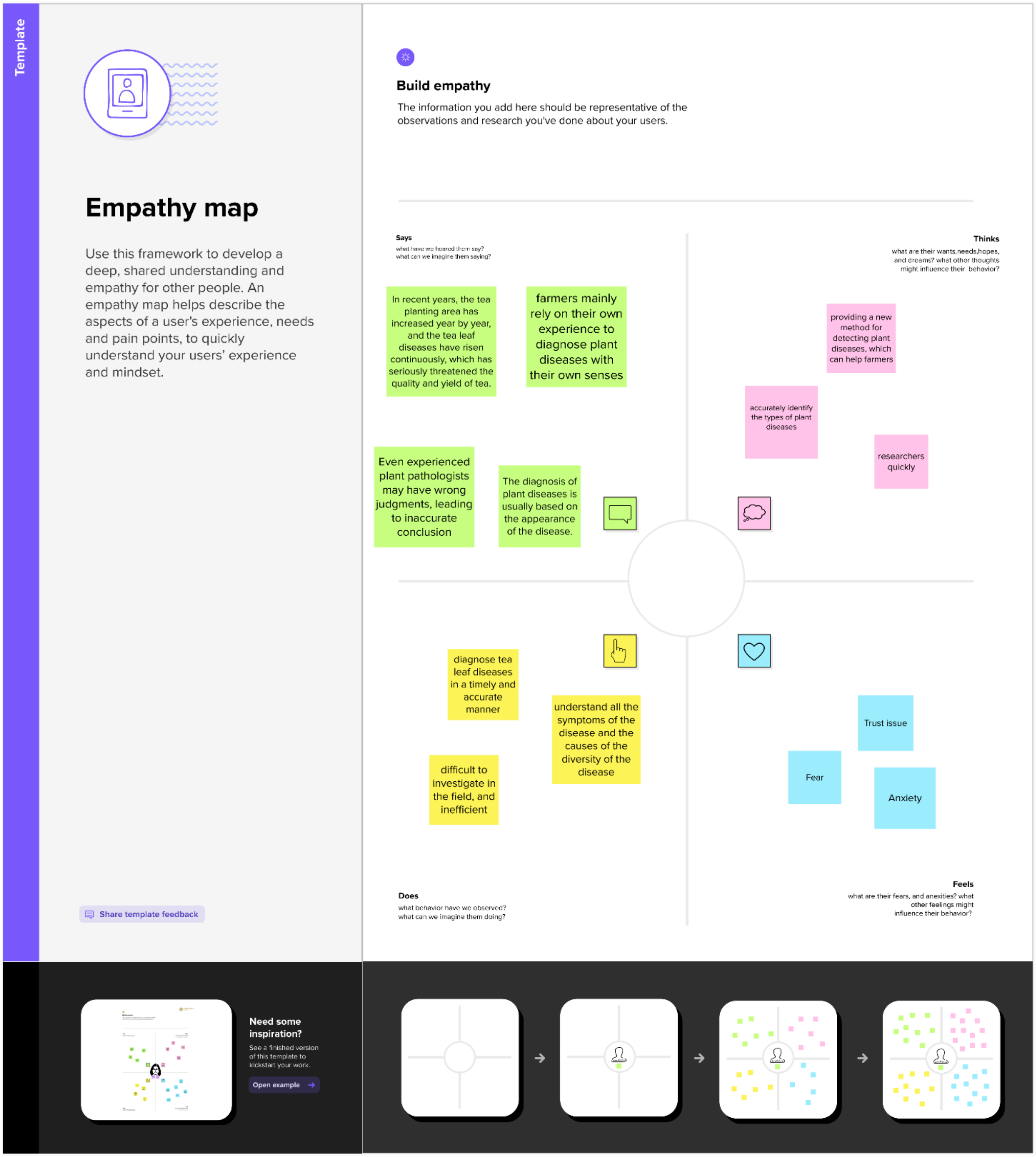
The purpose of developing a deep learning model for detecting diseases in tea leaves is to provide an automated and accurate solution for identifying and diagnosing diseases that affect tea plants. Tea plants can be susceptible to various diseases caused by pathogens, pests, or environmental factors, which can significantly impact tea production and quality.A deep learning model specifically designed for disease detection in tea leaves can analyze images or data collected from tea plantations and accurately classify the presence of diseases or abnormalities. This model can be trained on a large dataset of annotated tea leaf images, where each image is labeled with the corresponding disease or healthy state.The deep learning model learns to extract relevant features and patterns from the input data, enabling it to distinguish between healthy and diseased tea leaves. It can identify specific symptoms such as discoloration, spots, lesions, or other visual cues associated with various diseases. By leveraging the power of deep learning algorithms, the model can learn complex representations and generalize its understanding to new, unseen tea leaf images.

2.IDEATION & PROPOSED SOLUTION

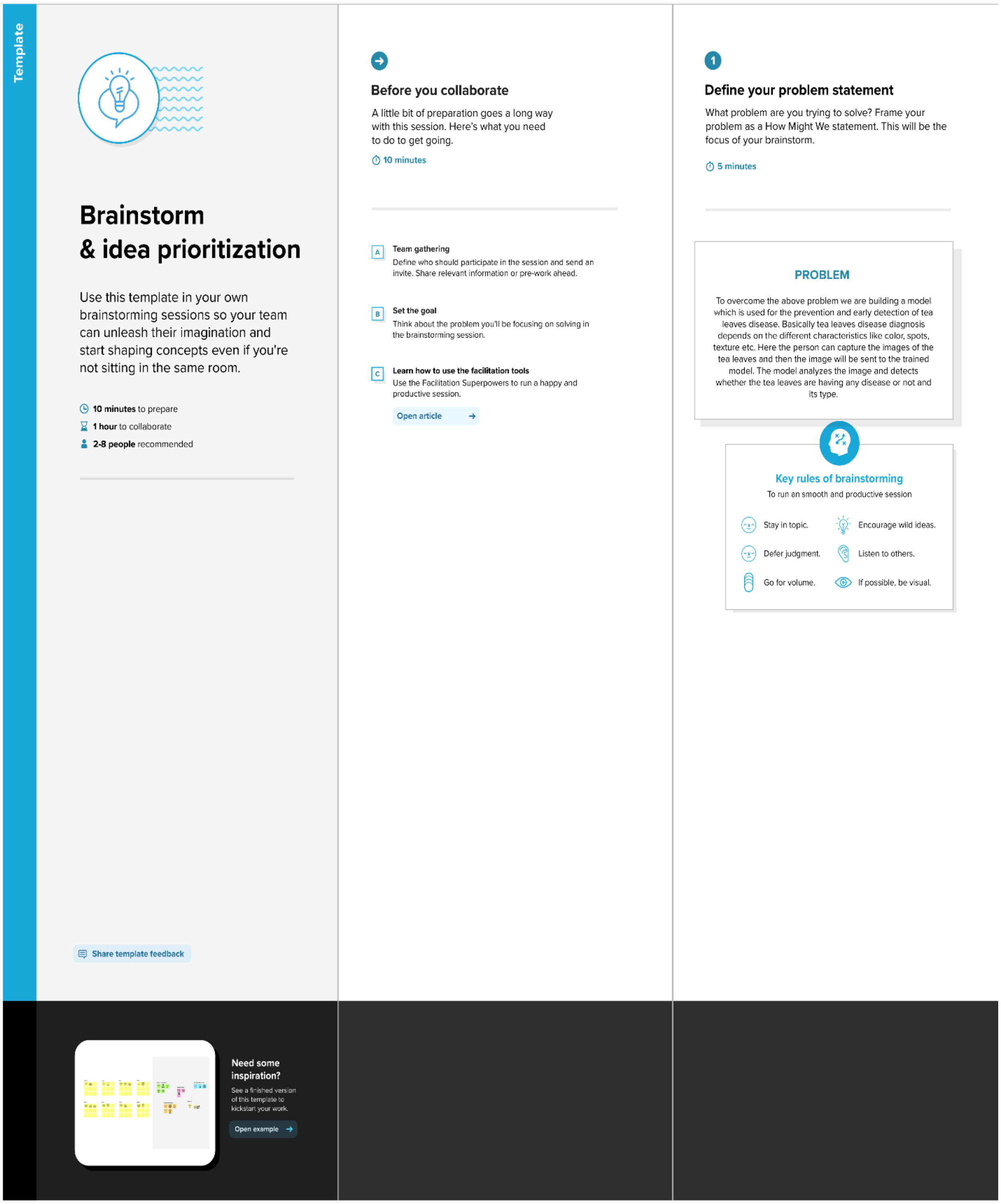
2.1. PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Gayathri	Get tea leaves tested for the detection of disease.	I am in difficulty.	I am afraid of being made a fool.	Irritable
PS-2	Nirmala	Get tea leaves tested for the detection of disease.	I hesitate.	I am afraid that wrong disease may be identified.	Fearful and anxious

2.2 EMPATHY MAP CANVAS



2.3 IDEATION AND BRAINSTORMING



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Strange spots & colors on the leaves may be an indication of disease. Experts and farmers can identify the type of disease by observing the leaves manually.

Tea leaf diseases can be identified by observing the leaves condition like color and spots on the leaves

Person 3

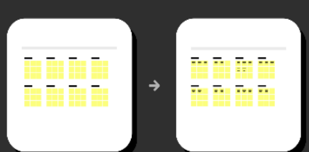
To overcome the above problem we are building a model which is used for the prevention and early detection of tea leaves disease. Basically tea leaves disease diagnosis depends on the different characteristics like color, spots, texture etc.

Here the person can capture the images of the tea leaves and then the image will be sent to the trained model. The model analyzes the image and detects whether the tea leaves are having any disease or not and its type.

Person 2

It is time-consuming and costly for experts to go to the tea garden for diagnosis

However, results are largely subjective when famers rely on their own experience to distinguish the types of tea diseases.



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

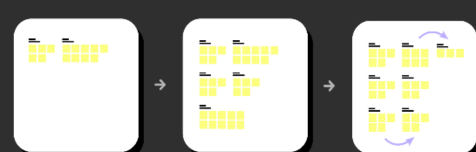
20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

At present, the diagnosis of tea leaf diseases relies on the manual method. Most tea trees grow in rugged mountainous areas. Thus, it is time-consuming and costly for experts to go to the tea garden for diagnosis. However, results are largely subjective when famers rely on their own experience to distinguish the types of tea diseases

Here the person can capture the images of the tea leaves and then the image will be sent to the trained model. The model analyzes the image and detects whether the tea leaves are having any disease or not and its type.



4

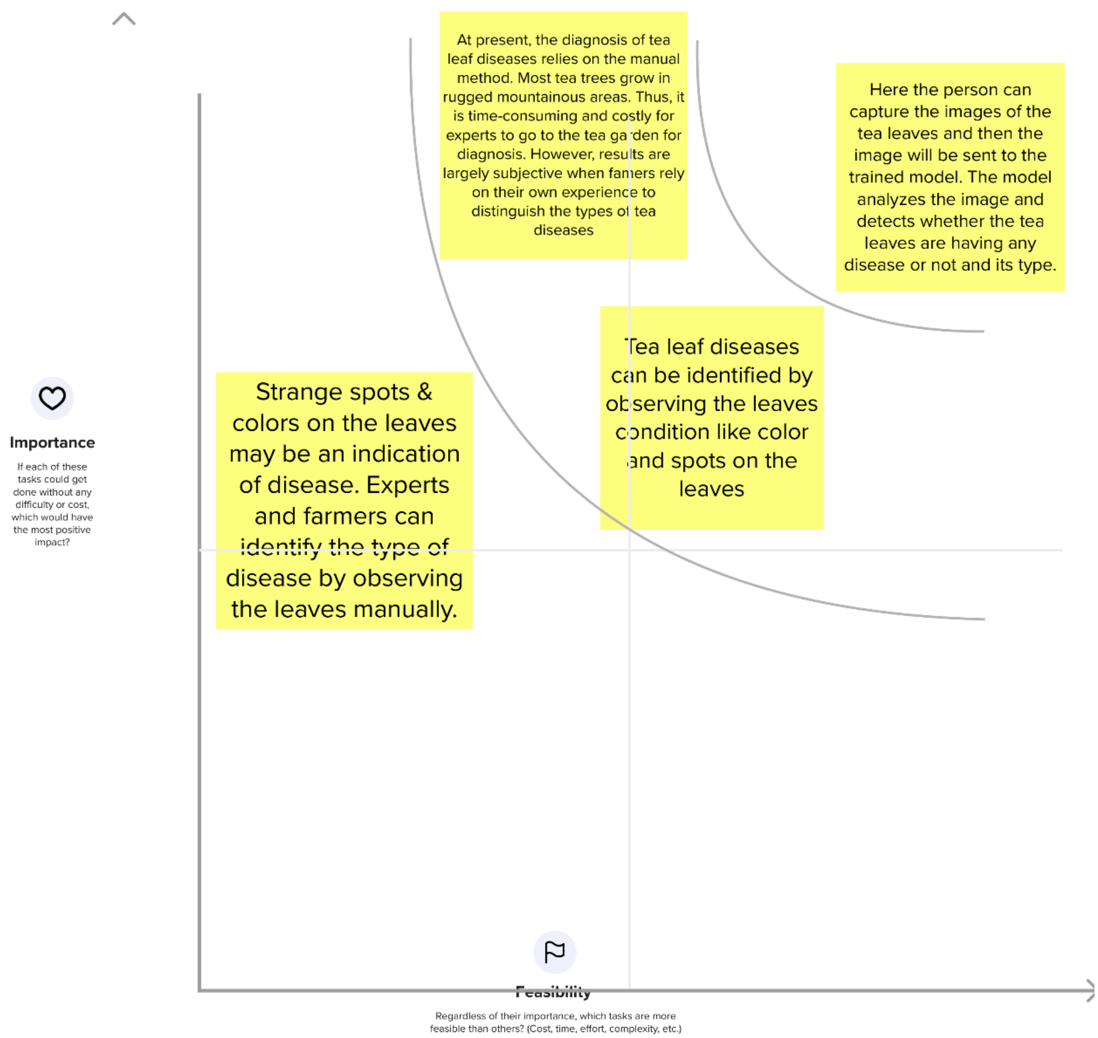
Prioritize

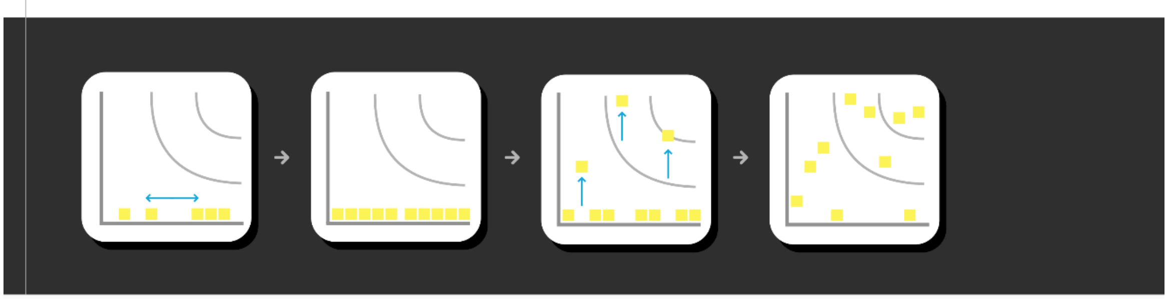
Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.





2.4. PROPOSED SOLUTION

S.NO	Parameter	Description
1	Problem statement (Problem to be solved)	Tea leaf diseases can also reduce the quality of tea and cause serious economic losses to tea farmers. Accurate detection and identification of tea leaf diseases and timely prevention and control measures are of great significance to reduce the loss of tea production, improve the quality of tea, and increase the income of tea farmers.
2	Idea / Solution description	To overcome the above problem, we are building a model which is used for the prevention and early detection of tea leaves disease. Basically tea leaves disease diagnosis depends on the different characteristics like colour, spots, texture etc. Here the person can capture the images of the tea leaves and then the image will be sent to the trained model. The model analyses the image and detects whether the tea leaves are having any disease or not and its type
3	Novelty / Uniqueness	This Project is expected to minimize the workload of experts and aid in rapid identification and detection of tea leaf diseases, thus minimizing economic losses.Therefore, it provides more precise andaccurate result.
4	Social Impact / Customer Satisfaction	It is less time consuming and low budget for the experts to identify the tea disease. It is more satisfactory for the experts to identify the tea leaf diseases
5	Business Model (Revenue Model)	We are associated with health centres, clinics, research centres and laboratories, so that the farmers and the experts get the tea leaf disease solution accordingly. Thus, through this AI project, we gain more profit by these organizations.
6	Scalability of the solution	As the tea leaf disease detection is done with the help of capturing the image of the tea leaves and detects whether the tea leaves are having any disease or not, so that this method is faster than the existing system,and it consumes low cost so the performance and scalability of this system is much high.

3.REQUIREMENT ANALYSIS

3.1. FUNCTIONAL REQUIREMENT

FR NO.	Functional Requirement (Epic)	Sub Requirements (Story / Sub-Task)
FR1	User Registration	A user registration system that allows users to create accounts and login credentials. The system should store user information securely, such as username, password, and email address.
FR2	User Confirmation	A user confirmation system that verifies the user's email address orphone number to ensure that the user is legitimate. This can involve sending a confirmation email or SMS to the user and requiring them to click a link or enter a code to confirm their account.
FR3	Accesscontrol	Access control mechanisms to ensure that only authenticated and authorized users canaccess the tea leaf disease detection system. This caninvolve role-based access control, where users are assigned different roles with different levels of access.
FR4	Data Privacy	Ensuring that user data and sensitive information, such as images of tea leaves, are stored and transmitted securely to protect user privacy.
FR5	User Feedback	A feedback mechanism that allows users to provide feedback

		on the accuracy and performance of the tea leaf disease detection system. This can be used to improve the system and provide users with a better experience
FR6	User Support	Providing user support to help users troubleshoot any issues they may encounter while using the system. This can involveproviding documentation, FAQs, and a help desk or ticketing system.

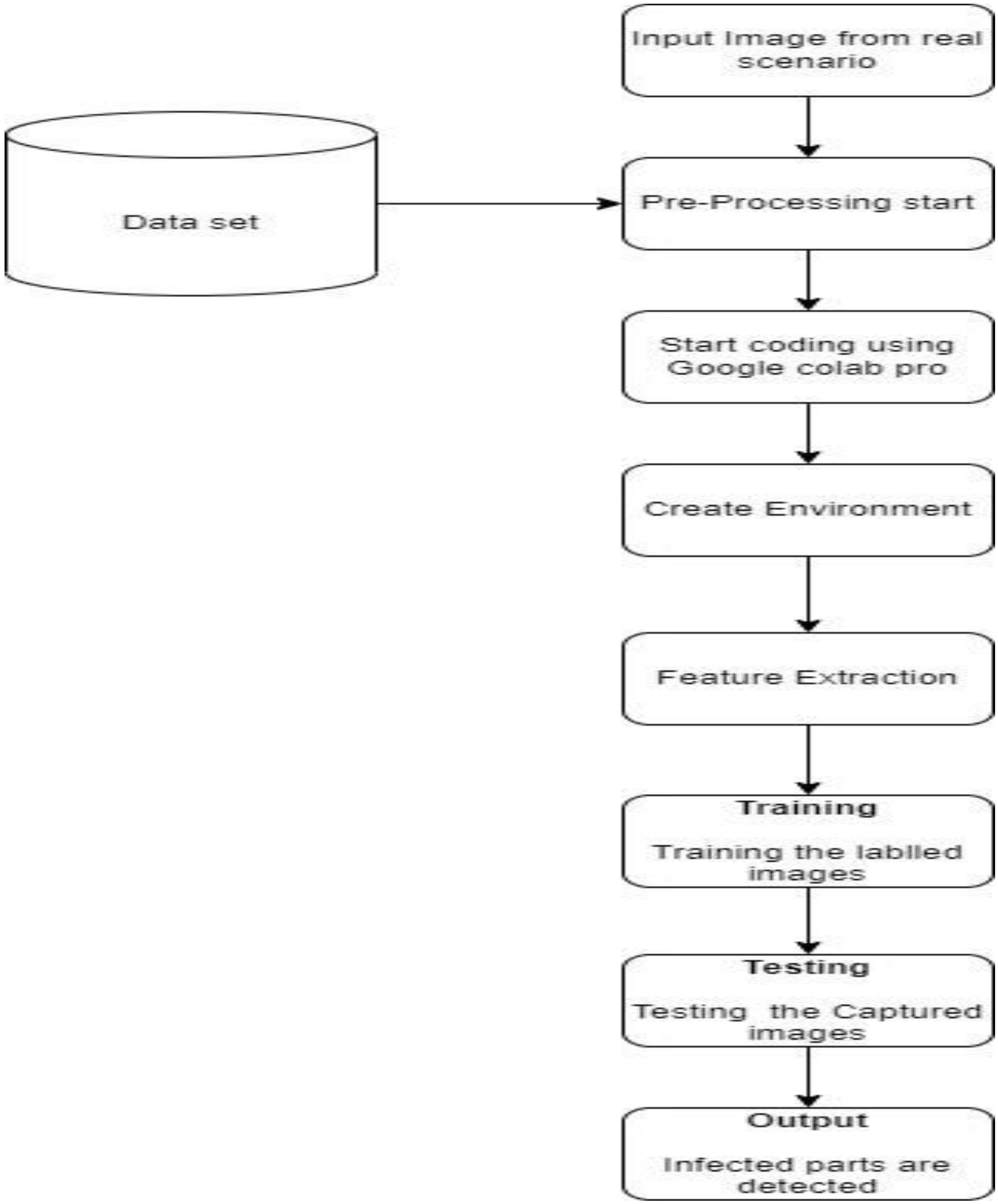
3.2 NON-FUNCTIONAL REQUIREMENTS

FRNO.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	The usability of a deep learning model for tea leaf disease detection involves its ease of use, flexibility, and user-friendliness. A user-friendly interface, easy-to-understand documentation, and comprehensive training resources can help to make the model more usable
NFR-2	**Security**	The security of a deep learning model for tea leaf disease detection is crucial as it may involve sensitive data. Adequate security measures, such as data encryption, access control, and regular security audits, can help to safeguard the model and prevent unauthorized access.
NFR-3	**Reliability**	The reliability of a deep learning model for tea leaf disease detection is vital for ensuring accurate and consistent results. It involves assessing the model's performance and identifying and fixing any issues that may arise. Regular testing and validation can help to maintain the model's reliability.

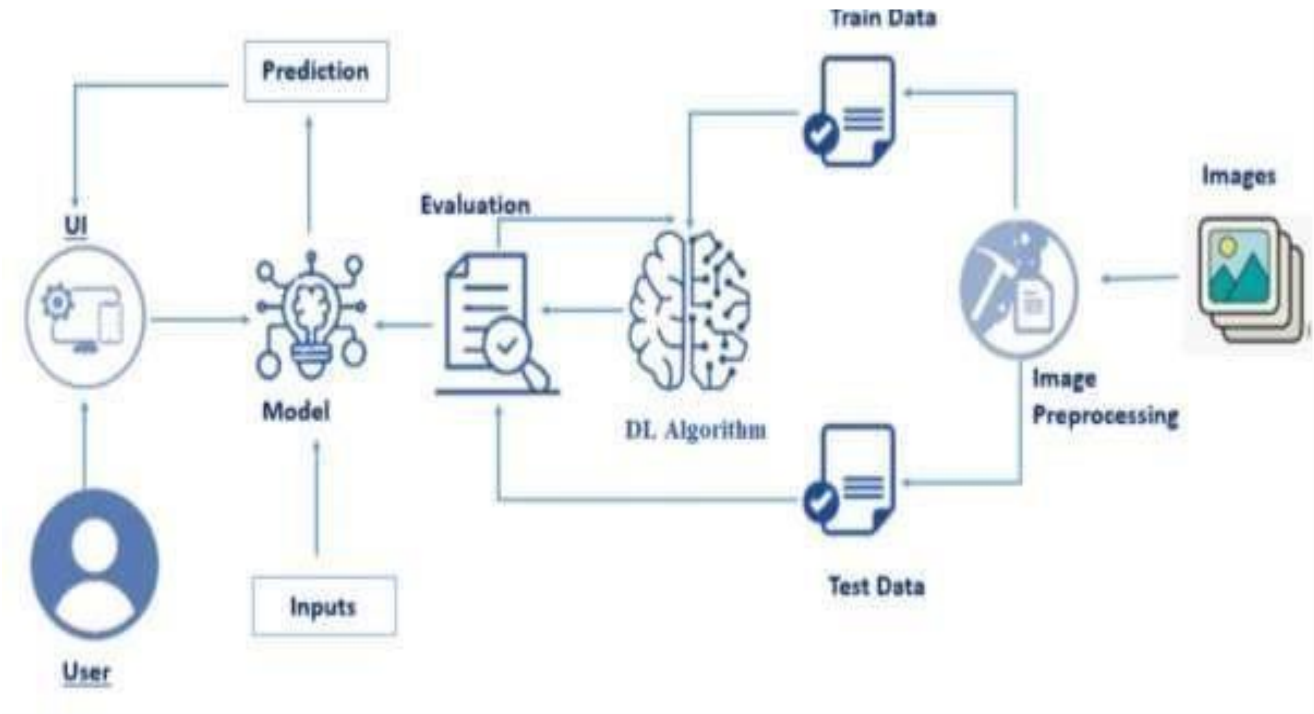
NFR-4	**Performance**	The performance of a deep learning model for tea leaf disease detection involves its speed, accuracy, and efficiency. The model should be able to accurately detect diseases in tea leaves within a reasonable amount of time, and with minimal false positives or negatives.
NFR-5	**Availability**	The availability of a deep learning model for tea leaf disease detection refers to its ability to be accessible and operational at all times. Measures such as redundancy, load balancing, and failover mechanisms can help to ensure high availability of the model.
NFR-6	**Scalability**	The scalability of a deep learning model for tea leaf disease detection refers to its ability to handle increasing amounts of data and users. As the volume of data increases, the model should be able to scale up or down to meet the demand. Horizontal scaling, vertical scaling, and distributed computing can help to ensure scalability.

4.PROJECT DESIGN

4.1. DATA FLOW DIAGRAM



4.2 SOLUTION AND TECHNICAL ARCHITECTURE



4.3. USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Member
Customer (Mobile user)	For capturing images using the device's camera.	USN-1	As a mobile user, I want to capture images of tea leaves using my device's camera, so that I can submit them for disease detection analysis.	The user can preview the captured image and choose to submit it for analysis.	High	Gayathri M

	The system should provide a file upload feature for submitting images.	USN-2	I want to upload images of tea leaves from my computer or mobile device, so that I can submit them for disease detection analysis.	The system successfully receives and processes the uploaded image for disease detection analysis.	High	Gayathri M
	The system should analyze the submitted images to identify disease symptoms or abnormalities accurately.	USN-3	I want the system to process the submitted images and identify any diseases or abnormalities present in the tea leaves.	The system successfully detects and identifies tea leaf diseases or abnormalities in the submitted images.	Low	Nirmala K
	The interface should provide clear instructions and guidance.	USN-4	I want to receive prompt and accurate results of the disease detection analysis.	Users can easily navigate through the system's interface without confusion or ambiguity.	Medium	Akshaya V

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**	**Priority**	**Team Member**
Customer (Web user)	A machine learning algorithm to analyze the data collected from the tea plants to detect any diseases and to display the notifications and the health status of the tea plants	USN-1	I want to be able to receive notifications of any diseases detected in my tea plants.	The machine learning algorithm accurately detects any diseases in the tea plants The user interface displays the notifications and the health status of the tea plants clearly and accurately	Medium	Akshaya V

Customer Care Executive	The system should be able to analyze the visual characteristics of tea leaves to detect potential diseases and provide suggestions for treatment.	USN-1	I want to be able to detect potential diseases in tea leaves so that I can provide accurate information and advice to customers.	The system should accurately detect at least 80% of the diseases present in tea leaves and a high level of accuracy and should be able to differentiate between different diseases with similar symptoms.	High	Gayathri M
Administrator	The digital tool should be able to provide a report on the type and severity of the disease detected and suggest appropriate measures to prevent the spread of the disease.	USN-1	I want to be able to detect diseases in tea leaves using a digital tool, so that I can take appropriate measures to prevent the spread of diseases and maintain the quality of tea production.	The digital tool should have a success rate of at least 90% in accurately detecting diseases in tea leaves and provide a detailed report on the type and severity of the disease detected within 2 minutes of analysis. It also measures to prevent the spread of the disease based on the type and severity of the disease detected.	High	Nirmala K

5.CODING AND SOLUTIONING

5.1 . Feature 1

Feature 1: Tea Leaf Disease Detection

Definition: Tea leaf disease detection refers to the process of using computer vision techniques and machine learning algorithms to automatically identify and classify diseases affecting tea leaves. This feature aims to assist tea farmers and researchers in monitoring the health of tea plants and detecting any signs of diseases at an early stage. By analyzing images of tea leaves, the system can provide timely and accurate information about the presence and severity of various diseases, enabling proactive measures to be taken to mitigate the damage and ensure the overall health of tea plantations.

Coding :

Here's a simplified coding example using Python and the OpenCV library for image processing and computer vision:

Python code

```
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

1.Data Preparation

```
# Assuming you have a dataset of tea leaf images labeled with different diseases.
# Preprocess the images and extract relevant features (e.g., color histograms, texture features).

# Load the dataset and labels
# ...

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(features, labels, test_size=0.2)
```

2.Training

```
# Train a machine learning model (e.g., Support Vector Machine) on the extracted features.
model = SVC()
model.fit(X_train, y_train)
```

3.Testing

```
# Evaluate the trained model on the testing set
y_pred = model.predict(X_test)
```

4. Performance Evaluation

```
# Assess the performance of the model using classification metrics
print(classification_report(y_test, y_pred))
```

5. Prediction

```
# Once the model is trained and evaluated, you can use it to predict the disease for new tea leaf images
def predict_disease(image):
    # Preprocess the input image
    # Extract features from the preprocessed image
    features = extract_features(image)
    # Use the trained model to predict the disease
    disease = model.predict([features])
    return disease
```

5.2.Feature 2

Definition: Real-time tea leaf disease detection refers to the capability of continuously monitoring tea plants and promptly detecting any signs of diseases as they occur. This feature involves using computer vision and machine learning algorithms to process live video or image streams from cameras installed in tea plantations. By analyzing the visual data in real-time, this feature enables farmers and researchers to quickly identify and respond to diseases, allowing for timely intervention and preventing further spread or damage to the tea plants.

Coding:

Here's a simplified coding example using Python and OpenCV for real-time tea leaf disease detection:

python code

```
import cv2
import numpy as np
from sklearn.externals import joblib

# Load the trained model
```

```
model = joblib.load('tea_leaf_model.pkl')

# Define the disease labels
labels = ['Healthy', 'Disease1', 'Disease2', 'Disease3']

# Define the color ranges for disease detection (adjust based on specific diseases)
color_ranges = [
    ((0, 0, 0), (255, 255, 255)),      # Healthy
    ((0, 0, 0), (50, 50, 50)),         # Disease1
    ((0, 50, 50), (50, 255, 255)),     # Disease2
    ((0, 100, 100), (50, 255, 255))    # Disease3
]

# Initialize the video capture from the camera
cap = cv2.VideoCapture(0)  # Use 0 for the default camera, or specify the camera index

while True:
    # Capture frame-by-frame
    ret, frame = cap.read()

    # Convert the frame to the HSV color space
    hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    # Detect diseases based on color ranges
    disease_detected = False
    for i, (lower, upper) in enumerate(color_ranges):
        lower = np.array(lower, dtype=np.uint8)
        upper = np.array(upper, dtype=np.uint8)
        mask = cv2.inRange(hsv, lower, upper)

        # Count the number of white pixels in the mask
        white_pixels = cv2.countNonZero(mask)

        # Check if the number of white pixels exceeds a threshold
        if white_pixels > 1000:  # Adjust the threshold based on specific conditions
            disease_detected = True
            disease_label = labels[i]
            break

    # Display the result on the frame
    if disease_detected:
        cv2.putText(frame, disease_label, (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0,
255), 2)
    else:
        cv2.putText(frame, 'Healthy', (10, 30), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

    # Display the frame
```

```
    cv2.imshow('Tea Leaf Disease Detection', frame)

    # Check for the 'q' key to exit the loop
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

# Release the capture and close all windows
cap.release()
cv2.destroyAllWindows()
```

6.RESULTS

6.1 Performance Metrics

- **Accuracy:**
  - Accuracy measures the proportion of correctly classified tea leaves (both diseased and healthy) by the model. It is calculated as the ratio of the number of correctly predicted samples to the total number of samples.

- **Precision:**

- Precision calculates the proportion of correctly predicted diseased tea leaves out of all tea leaves predicted as diseased by the model. Precision indicates how well the model avoids false positives, meaning correctly identifying healthy tea leaves as healthy.

- **Recall (Sensitivity or True Positive Rate):**
  - Recall measures the proportion of correctly predicted diseased tea leaves out of all actual diseased tea leaves. It indicates the model's ability to detect diseased samples and avoid false negatives, where diseased tea leaves are incorrectly identified as healthy.

- **F1 Score:**
  - The F1 score combines precision and recall into a single metric. It is the harmonic mean of precision and recall and provides a balanced evaluation of the model's performance. The F1 score is useful when there is an imbalance between the number of diseased and healthy tea leaves in the dataset.

- **Specificity (True Negative Rate):**
  - Specificity calculates the proportion of correctly predicted healthy tea leaves out of all actual healthy tea leaves. It indicates how well the model avoids false negatives, where healthy tea leaves are incorrectly identified as diseased.

- **Receiver Operating Characteristic (ROC) Curve:**
  - The ROC curve is a graphical representation of the model's performance across different classification thresholds. It plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) for various threshold values. The area under the ROC curve (AUC) provides a single-value metric summarizing the overall performance of the model. A higher AUC indicates better performance.

- **Confusion Matrix:**
  - The confusion matrix is a tabular representation that summarizes the model's predictions. It provides a breakdown of true positives, true negatives, false positives, and false negatives, allowing for a detailed analysis of the model's performance.

7.ADVANTAGES AND DISADVANTAGES

Advantages:

- High Accuracy:

  The models have demonstrated exceptional performance in various fields, including image recognition and disease detection. They can achieve high accuracy rates in detecting diseases in tea leaves, leading to more reliable diagnoses.

- Automated Detection:

It can automatically analyse large datasets of tea leaf images, enabling efficient and quick detection of diseases. This reduces the need for manual inspection and saves time and labour costs.

- Robustness to Variability:
  The capable of handling variations in tea leaf images caused by factors such as lighting conditions, camera angles, and leaf shapes. They can learn to extract meaningful features and patterns, making them more robust in identifying diseases under different circumstances.

- Scalability:
  It can be scaled up to handle large datasets, making them suitable for analysing extensive collections of tea leaf images. This scalability allows for the continuous improvement of the model's accuracy as more data becomes available.

Disadvantages:

- Data Requirements:
  It often requiresa substantial amount of labelled data to achieve optimal performance. Acquiring a sufficiently large and diverse dataset of tea leaf images with accurately labelled disease instances can be challenging and time-consuming.

- Interpretability:
  It considered black boxes because it is challenging to understand how they make predictions. This lack of interpretability can be problematic when attempting to explain the underlying factors contributing to disease detection in tea leaves.

- Computationally Intensive:
  The models can be computationally demanding, especially when dealing with complex architectures and large datasets. This can require significant computational resources, including powerful GPUs and substantial training times.

- Generalization Challenges:

It might struggle to generalize well to unseen data or adapt to new disease patterns. If the training data does not encompass the full spectrum of possible disease variations, the model's performance on real-world tea leaf images may be limited.

- Vulnerability to Noise:

  They can be sensitive to noisy or erroneous input data. In the case of tea leaf disease detection, image artifacts or irregularities may lead to misclassifications, reducing the model's overall accuracy.

8.CONCLUSION

In conclusion, deep learning models have shown great potential for detecting diseases in tea leaves. These models leverage the power of artificial neural networks to analyse large amounts of data and extract meaningful patterns that can be indicative of disease presence. By training these models on labelled datasets containing images of healthy and diseased tea leaves, they can learn to accurately classify and identify various diseases affecting tea plants.

The use of deep learning models for disease detection in tea leaves offers several advantages. Firstly, it provides a non-invasive and efficient method for early disease diagnosis, enabling prompt action to mitigate the spread and impact of diseases. This can potentially save time, resources, and crops by enabling targeted treatments or interventions. Secondly, these models can handle large-scale data analysis, making it possible to process a significant number of tea leaf images quickly, thereby increasing the efficiency of disease screening processes. Lastly, deep learning models have the potential for continuous improvement and adaptation through iterative training and retraining, allowing them to become more accurate and robust over time.

However, there are a few challenges that need to be addressed when implementing deep learning models for disease detection in tea leaves. One significant challenge is the need for extensive and diverse labeled datasets that cover a wide range of diseases, including both common and rare ones. Collecting and annotating such datasets can be time-consuming and require domain expertise. Additionally, there is a need to consider the interpretability of the deep learning models, as they are often regarded as black boxes. Efforts should be made to develop techniques that provide insights into the model's decision-making process, making it easier for experts to trust and validate the results.

Overall, deep learning models have the potential to revolutionize disease detection in tea leaves by offering accurate, efficient, and scalable solutions. Continued research and collaboration between experts in machine learning,deep learning, agriculture, and plant pathology will be crucial in further developing and refining these models to maximize their effectiveness in safeguarding tea plantations and ensuring a sustainable tea industry.

9.FUTURE SCOPE

- Dataset Expansion:
  - Deep learning models heavily rely on large and diverse datasets for effective training. Acquiring and annotating a more extensive

dataset of tea leaf diseases would improve the model's accuracy and generalization.

- Improved Accuracy:
  - Continued research and development can enhance the accuracy of deep learning models in disease detection. Techniques such as transfer learning, ensemble models, and advanced architectures like convolutional neural networks (CNNs) can be explored to achieve higher accuracy rates.

- Real-time Monitoring:
  - Integrating deep learning models with IoT (Internet of Things) devices can enable real-time monitoring of tea plantations. By deploying cameras or sensors in the fields, disease detection models can continuously analyse the health of tea leaves, providing timely alerts and allowing proactive disease management.

- Mobile Applications:
  - Developing user-friendly mobile applications that incorporate deep learning models can empower tea farmers and agronomists to easily detect and diagnose diseases in tea leaves. Such apps could provide instant feedback, disease identification, and recommendations for treatment.

- Automated Decision Support Systems:
  - It serves as the foundation for intelligent decision support systems in tea farming. By integrating disease detection models with other agricultural data, such as weather patterns, soil conditions, and historical crop performance, farmers can make informed decisions regarding disease prevention, crop management, and treatment strategies.

- Disease Identification and Classification:
  - It can be trained to identify and classify various diseases affecting tea leaves. This includes common diseases like blights, muds, leaf

spots, and viral infections. By accurately identifying the specific diseases, appropriate treatments can be applied, minimizing crop damage and yield loss.

- Disease Severity Assessment:
  - It can also be extended to assess the severity of tea leaf diseases. By analysing disease progression and symptoms, models can provide quantitative assessments of the damage inflicted, aiding in prioritizing treatment efforts and optimizing resource allocation.

- Generalization to Other Crops:
  - The deep learning models developed for tea leaf disease detection can potentially be generalized to other crops as well. By leveraging the knowledge gained from tea leaf disease detection, models can be adapted to detect and diagnose diseases in other plants, broadening their application and impact.

10.APPENDIX

SOURCE CODE

```
from tensorflow.keras.layers import Dense, Flatten, Input
from tensorflow.keras.models import Model
```

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.applications.vgg16 import VGG16, preprocess_input
import numpy as np
import matplotlib.pyplot as plt
imageSize = [224,224]
vgg = VGG16(input_shape=imageSize + [3], weights='imagenet',include_top=False)
for layer in vgg.layers:
    layer.trainable = False
x = Flatten()(vgg.output)
prediction = Dense(8, activation='softmax')(x)
model = Model(inputs=vgg.input, outputs = prediction)
model.summary()
model.compile(
loss='mse',
optimizer='adam',
metrics=['accuracy'], run_eagerly=True
)
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale= 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
training_set = train_datagenflow_from_directory('
c:\\Users\\indhu\\OneDrive\\Desktop\\tea_leaves\\dataset',
                                                 target_size = (224,224),
                                                 batch_size = 64,
                                                 class_mode = 'categorical')


test_set = test_datagen.flow_from_directory('
c:\\Users\\indhu\\OneDrive\\Desktop\\tea_leaves\\dataset',
                                            target_size = (224,224),
                                            batch_size = 32,
                                            class_mode = 'categorical')
```

```
training_set.class_indices
r=model.fit(
training_set,
validation_data=test_set,
epochs=20,
steps_per_epoch = len(training_set)//64,
validation_steps = len(test_set)//32
)
from tensorflow.keras.models import load_model
model.save('model_vgg.h5')
model=load_model('model_vgg.h5')
img = image.load_img("c:\\Users\\indhu\\OneDrive\\Desktop\\algal leaf.jpg",target_size=
(224,224))#loading of the image
x = image.img_to_array(img)#image to array
import numpy as np
x = np.expand_dims (x,axis = 0) #changing the shape
img_data = preprocess_input(x)
output = np.argmax(model.predict(img_data),axis=1)
index=['Anthracnose',
      'algal leaf',
      'brown blight',
      'gray light',
      'healthy',
      'red leaf spot',
      'white spot']
result=index[output[0]]
result
```

Github link: https://github.com/naanmudhalvan-SI/IBM--9632-1682399474

Video link:

https://drive.google.com/file/d/1ts1_GqbWzJtpYtbBoNtIuJLGkAMoGedS/view?usp=drivesdk