Mini Project Report on

PLANT DISEASE DETECTION USING MACHINE LEARNING

Submitted in partial fulfilment of the requirement for the award of thedegree of

BACHELOR OF

TECHNOLOGYIN

COMPUTER SCIENCE & ENGINEERING

Submitted by:

Student Name: Ranshul Kumar Jayant University Roll No.2021380

Under the Mentorship of

Mr. Kirit Joshi

Assistant Professor

Department of Computer Science and EngineeringGraphic Era (Deemed to be University)

Dehradun, Uttarakhand

January-2024

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"Plant disease detection using machine learning"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Kirit Joshi, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Name: Ranshul Kumar Jayant University Roll no: 2021380

Table of Contents

CHAPTER 1

INTRODUCTION

1.1 INRODUCTION:

The agriculture is one of the main pillars of our country and economy. About 58% of Indian population depends on the agriculture and it vary state vise. As Indian agriculture sector is more dependent on traditional practices and do not include much use of technology in this domain which stops them to compete at the international level. The major issue that Indian farmers face is detection of plant disease at early stage and protect their crop from these diseases.

1.2 AIM:

The aim of the project is to focus on one of the important problems of the agriculture which is the disease detection. The disease in the plant or crop affects the production and economy, so the model is introduced to automate the process and easily identify the diseases that are affecting plants and raised the production of the farmers and save them from the losses they suffer.

1.3 PROBLEM STATEMENT:

Plant disease had been the major problem faced by the farmers in agriculture and manually detecting plant disease using the traditional method require prior knowledge about the disease, tremendous quantity of labor and the excess time to check plants and detect the diseases. In this project, we are working on the plant disease detection using machine learning. The potato plant is used to detect the diseases in the potato plant using machine learning. The farmers who grow potato are facing lots od economical losses every year because of the diseases that can happen to a potato plant. There are two common diseases in potato plants known as Early Blight and Late blight. Early Blight is caused by fungus and Late Blight is caused by oomycete pathogen Phytophthora infectants. If farmer can detect these problems early and apply appropriate treatment then it can save lot of waste and prevent the economic loss. Machine learning is the branch of artificial intelligence which works automatically after training it. The main aim of this project is to train the machine learning model using the images of the infected plant and healthy plant which can help the farmers to grow their business and detection of disease at early stage.

1.4 DATA COLLECTION:

We have used the Kaggle dataset for the collection of data for this project. The name of the dataset is PlantVillage in which we have used he images of the potato plant leaf. The dataset contains the 944 images each of the Early Blight and Late Blight disease, around 152 images of the healthy plant leaves. The total collection of datasets is around 2020 images. The machine learning model is trained on these datasets on 80-20 basis in which 80% data is used as a training data and in remaining 20% data, 10% is used for validation and 10% testing of machine learning model.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

[1] Amrita S.Tulshan and Nataasha Raul has presented that the utilization of machine learning techniques for plant disease detection, focusing on the agricultural domain. The study explores the effectiveness of various algorithms, including k-Nearest Neighbour (KNN), Probabilistic Neural Network, Genetic Algorithm, Support Vector Machine (SVM), and Principal Component Analysis, in classifying and diagnosing plant diseases based on image features. The paper emphasizes advancements in image processing techniques, particularly the Gray Level Co-occurrence Matrix (GLCM) for texture analysis. It also addresses challenges related to dataset diversity and model generalization, proposing solutions such as transfer learning and ensemble methods. The research paper conducts a comparative analysis, showcasing the superiority of the proposed KNN classifier over the existing Linear SVM classifier in terms of accuracy (98.56% vs. 97.6%). This literature survey provides a concise overview of the evolving landscape of machine learning applications in plant disease detection, highlighting the need for continuous improvement in algorithms and methodologies to enhance accuracy and robustness in real-world agricultural scenarios.

Conclusion:

This highlights the evolution of machine learning applications in plant disease detection, emphasizing the diversity of algorithms and image processing techniques employed. It sets the stage for the proposed work by providing a comprehensive understanding of existing methodologies and paving the way for advancements in accurate and timely disease identification.

Significance and Potential Applications of this model:

This study illustrates the potential of machine learning in revolutionizing plant disease detection, particularly for vital crops like potatoes. The ability to accurately identify diseases like Early and Late Blight can significantly contribute to preventative measures in agriculture, enhancing crop yields and sustainability.

[2] In addressing the significant challenge of crop disease identification for global food security, the paper by Ramesh et al. introduces a novel approach employing Random Forest for the classification of healthy and diseased leaves through leaf-based image analysis. The research encompasses crucial stages including dataset creation, feature extraction using Histogram of Oriented Gradient (HOG), classifier training, and classification. By leveraging machine learning on publicly available datasets, the study aims to offer a scalable solution for large-scale plant disease detection. The literature review acknowledges the emergence of diverse technologies and methodologies in disease identification, encompassing traditional machine learning methods such as random forest, artificial neural networks, support vector machines, fuzzy logic, and convolutional neural networks. Notable works in this domain are reviewed, presenting various techniques like Back Propagation Neural Network, Pattern Recognition Techniques, Computer Vision Technology, and Multiple Classifier Systems. The proposed methodology integrates three key feature descriptors - Hu moments, Hara lick Texture, and Colour Histogram - contributing to a comprehensive analysis of leaf images. The research utilizes Random Forest as a flexible classification technique, addressing overfitting issues prevalent in decision trees. The literature concludes with a discussion on the algorithm's efficacy, highlighting its comparison with other machine learning models and indicating Random Forest's superior accuracy (70.14%) in classifying diseased and healthy leaves compared to alternative models. Overall, the literature survey provides a robust overview of the application of machine learning in plant disease detection, emphasizing the potential of Random Forest for accurate and scalable solutions in agricultural settings.

CHAPTER 3

METHODOLOGY

3.1 DATASET VISUALISTION

The dataset include 3 folders which is explained above -:

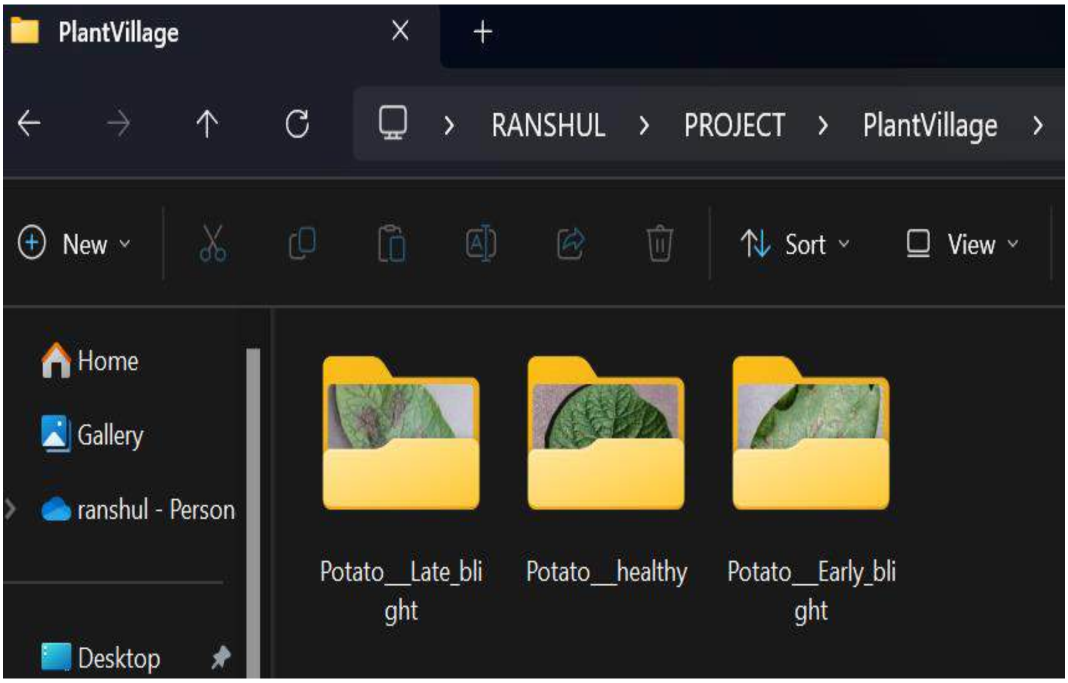


Fig 3.1. Datasets

Now some samples of the datasets are shown in fig. 2.

3.1.1 Matplotlib

Matplotlib is a python library which is used in the visualization of image in this and all the images are in the form of the 3D array and RGB which is further converted into the NumPy array.

```
for image_batch, labels_batch in dataset.take(1):
    print(image_batch.shape)
    print(labels_batch.numpy())
```

```
(32, 256, 256, 3)
[0 0 0 0 1 1 0 0 1 1 2 2 0 1 0 1 0 1 0 1 1 0 0 0 1 0 1 1 1 0 1 0]
```

Fig 3.2 Image representation

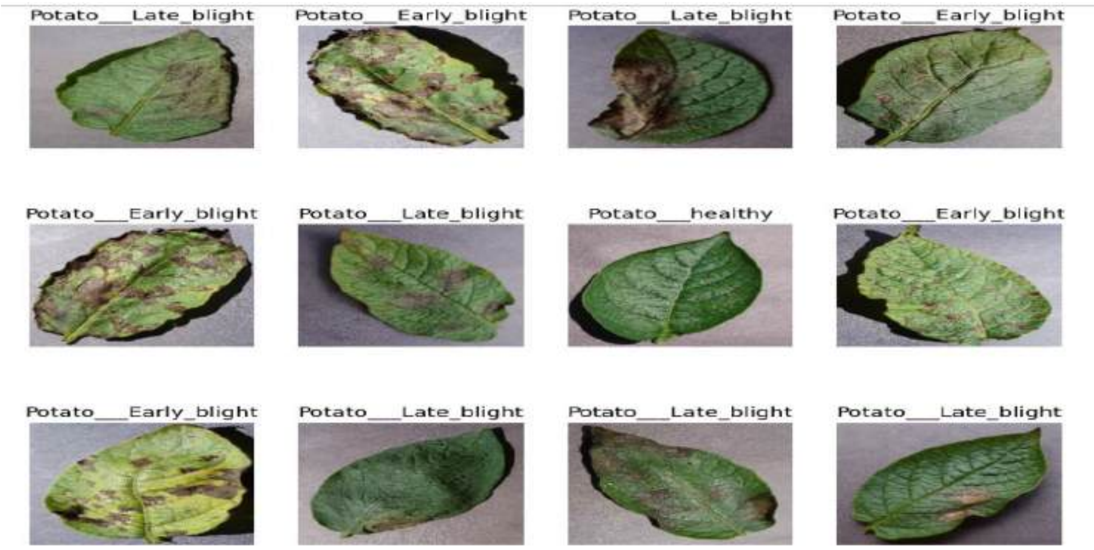


Fig 3.3 Some samples of the dataset.

3.2 DATA CLEANING AND PREPROCESSING

3.2.1 TensorFlow

TensorFlow is the open-source library for the Machine Learning and Artificial Intelligence. It is developed by the google for the AI Engineers and it is used for the Neural Network image recognition. Data cleaning is done using the tf_dataset which divides the images into the number of batches and removing the blurry image to enhance the quality of the dataset.

```
BATCH_SIZE = 32
IMAGE_SIZE = 256
CHANNELS=3
EPOCHS=25
```

```
len(dataset)
```

```
[7]:

64
```

Fig 3.4 TF dataset.

So, the batch size is taken as 32 and images are divided into the 64 batches. TF dataset is the data structure which is used to classify millions of images into the batches so they can work perfectly.

TensorFlow pipeline is used to improve the performance and accuracy of the TensorFlow model. Pipeline is used for data preprocessing. It is used to handle large datasets and we can apply various transformation.

3.2.2 Prefetch and cache is used to improve the performance of the model. Prefetch will load the new batch of data from the memory while GPU is busy training the previous batch. Catch will used to not read the same image again.

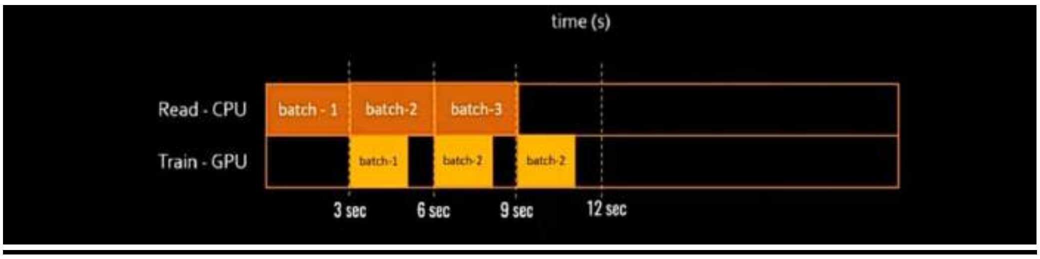


Fig 3.5 Prefetch

3.2.3 DATA AUGMENTATION – When we train a model and after that when we supply rotated image then it will not work better. Data augmentation is used to make our model robust. If we supply rotated image to our model so it will not confuse. The image is rotated in every possible case and train the model to all the condition.

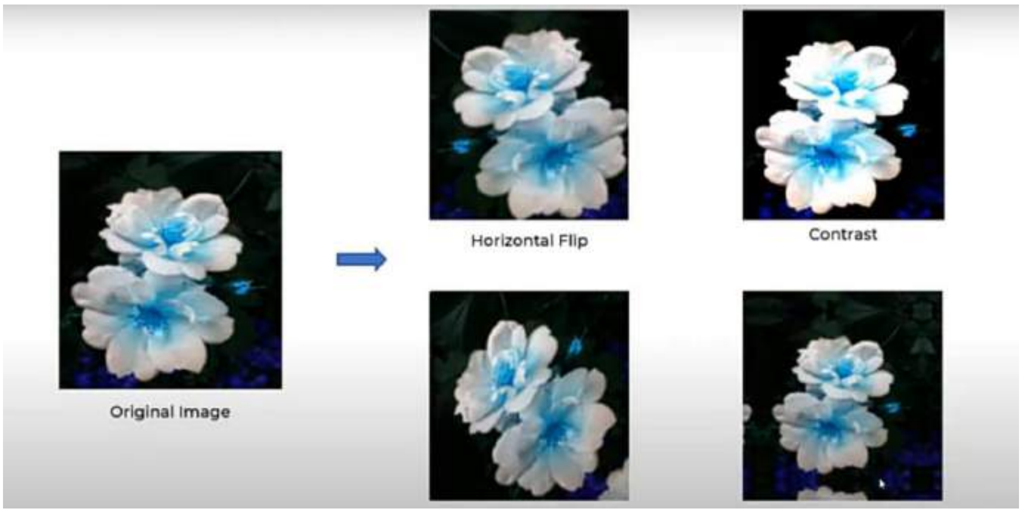


Fig 3.6 Data Augmentation

3.3 MODEL BUILDING ALGORITHM

CNN (CONVOLUTION NEURAL NETWORK) is used to build the network and we train the CNN Model on our train data dataset (80% of total data for training). After training the model we will test it on the test data set which is 10% of the dataset and calculate the accuracy of the model.

Feature map is used to detect the feature of the object. Kernel size is filtering size which is 3.3. Feature map is then forming the 3D object the make the detection of the specific part. 3D objects are then flattened and converted into the 1D array and join them together and we will get the full neural network.

A common CNN model architecture is to have a number of convolution and pooling layers stacked one after the other. Pooling is used to reduce the dimension of the feature map. Max pooling is the most effective one in which largest numbers form 2*2 matrix and reduce the computation. Accuracy is the metric and Adam is the optimizer.

```
input_shape = (BATCH_SIZE, IMAGE_SIZE, IMAGE_SIZE, CHANNELS)
n_classes = 3

model = models.Sequential([
    resize_and_rescale,
    layers.Conv2D(32, kernel_size = (3,3), activation='relu', input_shape=input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64,  kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(32, (3, 3), activation='sigmoid'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(32, (3, 3), activation='sigmoid'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(n_classes, activation='softmax'),
])

model.build(input_shape=input_shape)
```

```
model.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
    metrics=['accuracy']
)
```

[47]:

```
history = model.fit(
    train_ds,
    batch_size=BATCH_SIZE,
    validation_data=val_ds,
    verbose=1,
    epochs=25,
)
```

```
Epoch 1/25
51/51 [==============================] - 48s 903ms/step - loss: 0.9332 - accuracy: 0.463
8 - val_loss: 0.8973 - val_accuracy: 0.4792
Epoch 2/25
51/51 [==============================] - 44s 853ms/step - loss: 0.7843 - accuracy: 0.628
1 - val_loss: 0.5862 - val_accuracy: 0.8021
Epoch 3/25
51/51 [==============================] - 44s 855ms/step - loss: 0.5844 - accuracy: 0.776
3 - val_loss: 0.7008 - val_accuracy: 0.6979
Epoch 4/25
51/51 [==============================] - 45s 876ms/step - loss: 0.5141 - accuracy: 0.807
6 - val_loss: 0.6390 - val_accuracy: 0.7448
Epoch 5/25
51/51 [==============================] - 44s 856ms/step - loss: 0.4590 - accuracy: 0.841
```

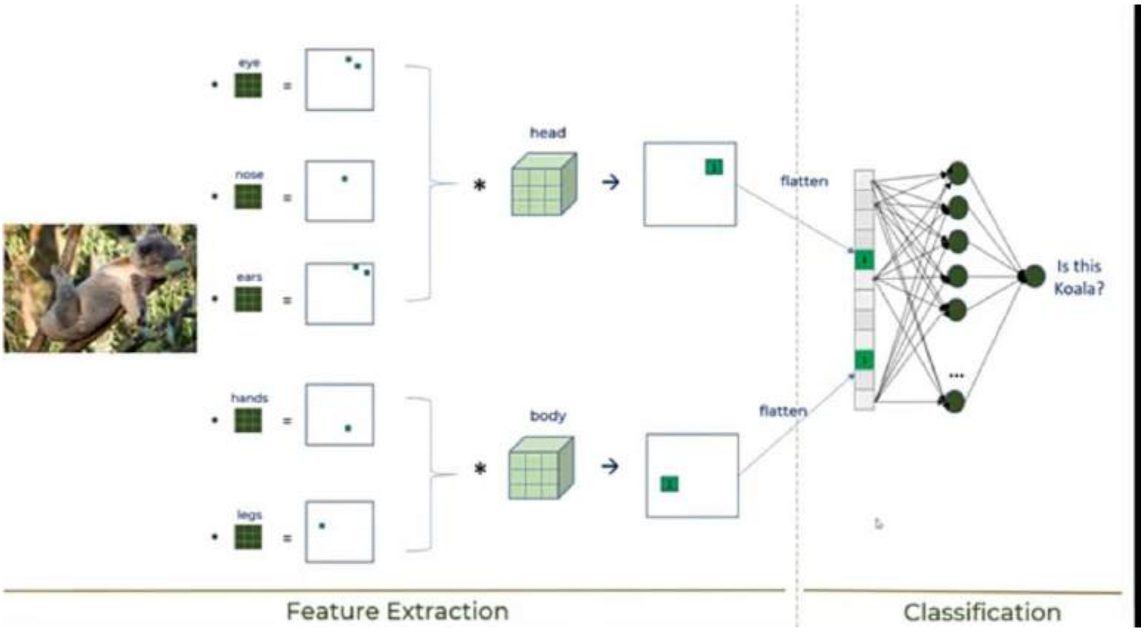
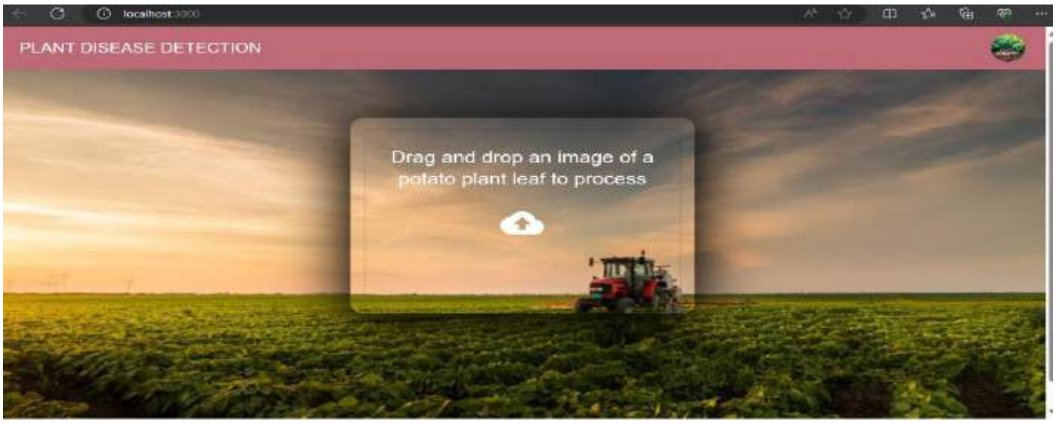


Fig 3.7 CNN

3.4 FRONTEND AND BACKEND:

A website is made using the React Js. React is the library developed by the facebook to get data in the real time . It is used to reusability of the components and do not reload the page to update the data to the website . For desinging CSS is used .

The frontend is connected to the backend which is made up of using FastApi. FastApi is the modern web framework to build the api's in pyhton to write a server . It offes inbuild data validation and documentation support. It is very fast and compacr with very less bugs.



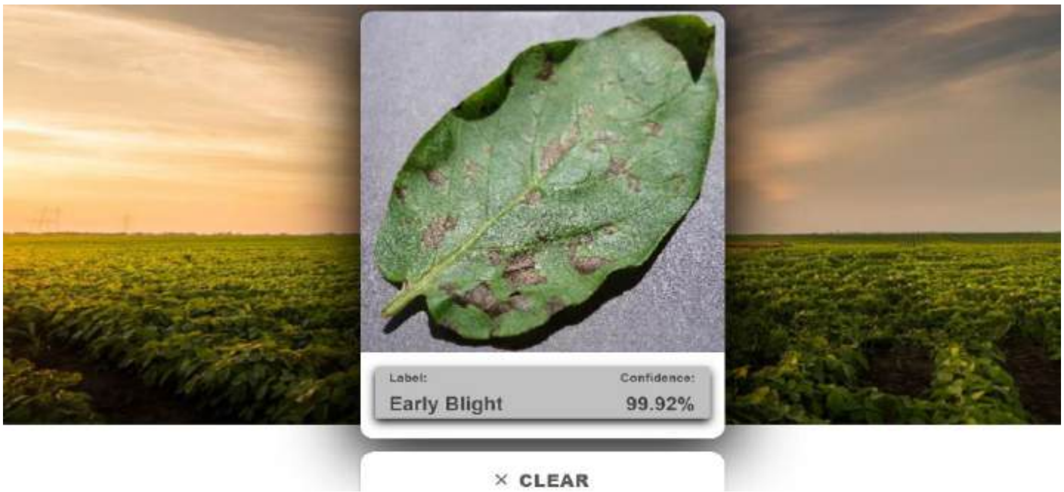


Fig 3.8 Frontend of the website detecting diseases

CHAPTER 4

RESULT AND DISSCUSION

In this section, we present a detailed analysis of the outcomes obtained from our machine learning models for plant disease detection. Our primary objective was to design and implement effective models capable detecting plant disease effectively. The models were trained and evaluated using a comprehensive dataset, employing CNN algorithms.

4.1 ACCURACY -:

THE OVERALL CORRECTNESS OF OUR MODEL'S PREDICTION IS CALLED THE ACCURACY.

FORMULA = (TOTAL CORRECT PREDICTION /TOTAL PREDICTION)

```
scores= model.evaluate(test_ds)
```

```
7/7 [==============================] - 2s 272ms/step - loss: 0.1791 - accuracy: 0.9375
```

Fig 4.1 ACCURACY

4.2 ACCURACY AND LOSS GRAPH

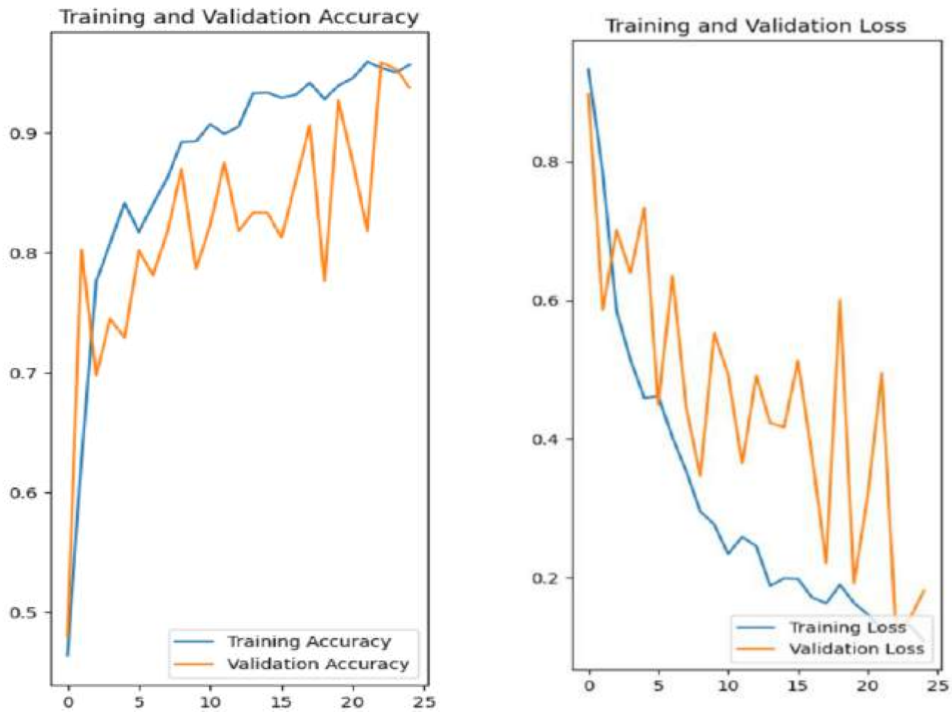


Fig 4.2 Accuracy and Loss Graph

[55]:

```
acc=history.history['accuracy']
val_acc= history.history['val_accuracy']

loss=history.history['loss']
val_loss=history.history['val_loss']
```

[58]:

```
plt.figure(figsize=(8,8))
plt.subplot(1,2,1)
plt.plot(range(EPOCHS),acc,label='Training Accuracy')
plt.plot(range(EPOCHS),val_acc,label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.figure(figsize=(8,8))
plt.subplot(1,2,1)
plt.plot(range(EPOCHS),loss,label='Training Loss')
plt.plot(range(EPOCHS),val_loss,label='Validation Loss')
plt.legend(loc='lower right')
plt.title('Training and Validation Loss')
```

4.3 Challenges Faced -: The model performance is contingent on the quality of data its trained on. In the scenarios where dataset lacks diversity or fails to evolving trends, the model may face challenges .

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION:

The project use machine learning technique to give results, with CNN (CONVOLUTION NEURAL NETWORK). The CNN algorithm provides good result. As result, it is shown that current model is working properly and giving good result to the provided inputs. The above model will work for the betterment of the famers of the country. It will reduce the farmers income loss and help them to safeguard their crops from the disease using the model and detecting the diseases at early stage. The model can be improved by training it more with the larger dataset which will make the model to predict with more accuracy.

5.2 FUTURE WORK:

Moving forward, there are several avenues for the future work that could enhance the capability of the model and increase the predictability:

5.2.1 Datasets:

Training the model to more datasets increase the efficiency of the model.

5.2.2 ALGORITHM REFINEMENT:

The advanced machine learning algorithm can be explored to boosts the model efficiency and model can performed much better.

In the future, we plan to employ this approach for the examination of diverse datasets originating from various countries. Our devised system, designed for generalization, will facilitate the analysis of multiple class labels during the prediction phase. This has the potential to significantly enhance the efficiency of crop yield and opens up an avenue for promising future research.

REFERANCES

[1] Tulshan, Amrita S., and Nataasha Raul. "Plant leaf disease detection using machine learning." *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019. **(Research paper)**

[2] S. Ramesh et al., "Plant Disease Detection Using Machine Learning," 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, India, 2018, pp. 41-45, doi: 10.1109/ICDI3C.2018.00017. **(Research paper)**

[3] D. Gosai, B. Kaka, D. Garg, R. Patel and A. Ganatra, "Plant Disease Detection and Classification Using Machine Learning Algorithm," 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-6, doi: 10.1109/ICONAT53423.2022.9726036.