**Department of Electrical and Computer Engineering**

**North South University**

**Junior Design Project**

# Leaf Disease Detection Using Machine Learning

MIA MAHMADUL ISLAM SHAON; ID: 2014314042

ASIF RAHMAN; ID: 2012691042

MD. NAZMUL ISLAM SRABON; ID: 2021227042

**Section: 02, Group: 05**

## Faculty Advisor:

Dr. Tanzilur Rahman

Department of ECE

Spring 2023

# ABSTRACT

Rooftop gardening is considered as one of the most popular and emerging hobby recently in this busy over polluted city. rooftop gardening is considered as one of the most popular and emerging hobby recently in this busy over polluted city there is hardly any space for agriculture. Thus people fulfil their desire in their respective rooftops. However people also loses interest in it sooner or later. Researchers have found out why people get rid of these amazing hobby. The answer is simple their plant starts to perish after they get exposed to some disease and the owners cannot do anything. Recently research also proved that plant diseases are detected by leaf disease thus detection of plant disease through some automatic technique is the ultimate necessity is it reduces hassle free detection with remedies recommending simple remedies. This project presents a machine that is made by algorithm for image segmentation which is used to detect and classify plant leaf diseases. Our algorithm is used to segment images which is crucial step in the disease identification process for leaf diseases

# Table of Contents

# CHAPTER 1: INTRODUCTION

In this chapter, we are going to discuss about rooftop gardening. The impact of leaf diseases on rooftop gardening, why the detection of leaf disease is necessary, our project's aim, objectives, and the motivation for doing this project.

## 1.1 Rooftop Gardening

In this modern day and era, people of all classes try to incorporate the nature with their lives in various ways. Gardening is one of the most prominent of ways to do that. Due to the lack of open space, urban people have leaned towards rooftop gardening. Rooftop gardening refers to the practice of growing plants and vegetables on the rooftop of a building. This practice has emerged as a number one hobby for most of the urban people in recent years.

Rooftop gardening has several important benefits which includes improved air quality, increased access to fresh produce. Since the presence of green plants on the rooftop helps to absorb the increased heat due to heat absorbing materials such as concrete and asphalt, it is known to reduce the urban heat island effect. **[1]**

Overall, rooftop gardening can be considered as a great way of utilizing the limited urban space while adding several environmental, social and economic benefits.

### 1.1.1 Major Obstacle of Roof Gardening: Leaf Disease

Leaf disease is one of the major obstacles in rooftop gardening. Leaf disease can be considered as a small, discolored or diseased patch of a leaf that is often brought on by nematodes, insects, environmental conditions, toxicity, or herbicides. It can also be brought on by fungal, bacterial, or viral plant diseases. These tan patches or lesions frequently have a necrotic center. **[2]** Because of their exposure to wind, temperature changes, and other environmental stressors, rooftop gardens are particularly prone to leaf disease. So, it's critical to frequently check the garden for disease-related symptoms and respond quickly to stop any further spread. But, most of the urban people lack the required time and knowledge of farming. Hence, they are often unable to detect leaf diseases in time which causes the plant to perish. Ultimately, most of the urban people lose the enthusiasm for roof gardening. To remove this obstacle, we are proposing a machine learning based technological tool that can make the leaf disease detection process much easier and less time consuming.

## 1.2 Machine Learning

The fact that computers and other machines are rigid logic machines with no common sense at all means that they are required to have some instructions in order to do tasks. So, to complete a task, precise and step-by-step instructions must be given to the machine. Computers are programmed to follow and carry out the instructions, and scripts are afterwards produced to do so. That is where machine learning, the concept of training computers or other machines based on the experiences from previous data, comes in. Machine learning (ML) is a topic of study, which focuses on comprehending and developing "learning" methods, methods that use data to enhance a set of tasks. **[3]** It is a component of artificial intelligence. Machine learning uses a variety of techniques to train the computers to carry out some tasks for which there isn't a totally suitable solution. To do that one can assign some of the right answers as valid when there are many possible replies. This valid data can be used by the computer as the practice data to refine the algorithm's it employs to determine the right answers. For instance, the MNIST dataset of handwritten digits has frequently been used to train a system for the task of digital character recognition. **[4]**

## 1.3 Project Aim and Objective

In this modern age of technology, one would presume that technology has touched all spheres of our life. Even though it is true to some extent. But the use of technology in detecting the leaf disease is still not a common practice. Most of the people still rely on professionals entirely. Hence, the disease detection process is often time-consuming, and the accuracy of diagnosis depends on the operator's concentration level and mental state. This problem could easily make the gardener lose enthusiasm about gardening altogether. But upon achieving our aims and objectives about this project, we hope to minimize the complications to a certain degree. The aim of our project is to develop a mobile application that can detect leaf disease in real time and also suggest suitable remedies. The points below are our project's objective:

- To find an optimized dataset to build our model
- To find an optimized way for image pre-processing
- To build an easily implementable ML model for the real world
- To assist the urban people in taking necessary steps without wasting a lot of time

## 1.4 Motivation

The motivation of this project is totally based on the target user of gardening and the severity of the leaf disease. It is quite clear that gardening has become a common practice for urban life. Nowadays, Especially in Dhaka city we can hardly find a roof that has no plant in it. But still, we cannot neglect the fact that most of the urban people involved in gardening has no prior knowledge about farming. They do not know what are the diseases the plants can face, how to detect these diseases and finally how to remedy them. Hence, they often fail to detect the disease in time. Also, leaf disease can be both infectious and non-infectious depending on the nature of a causative agent. If not detected early it can lead to the death of the plant which could have easily been solved by a touch of technology. But most of the features of agricultural technology focuses mainly on field agricultural crops rather that residential gardening. The reason why we are taking this project is to help these urban people get acquainted with a technological tool that will help them to detect the plant diseases in real time.

# CHAPTER 2: LITERATURE STUDY

To get some ideas about our project we researched some of the existing works related to our project. In this chapter we are going to discuss about these literature survey.

## 2.1 Existing Works Explanation

A paper has been created in **[5]** by Sneha Patel about leaf disease detection using convolutional neural network. The main aim of this paper is to detect the apple, grape, corn, potato, and tomato plants leaf disease. The paper proposed Deep CNN model as the solution. Also the proposed CNN model has then been compared with popular transfer learning approach such as VGG16. This paper has followed some steps like image acquisition, image pre-processing, image segmentation, features extraction and classification. The paper used a dataset containing 61,486 images. They used six different augmentation techniques for increasing the data-set size. These techniques are image flipping, gamma correction, noise injection, PCA color augmentation, rotation and scaling. It divided the dataset in 24 labels according to the diseases. The tools used by this paper are Python, Numpy, Scikit learn, Tensorflow, Keras, Compiler option and Jupiter notebook. The proposed CNN model has an accuracy of 90.23%. Whereas the VGG16 model has an accuracy of only 51.17%. So Deep CNN model has better result compared to VGG16.

A study has been done **[6]** by Vijai Singh and A.K. Misra to detect plant leaf diseases using image segmentation and soft computing techniques to categorize plant leaf diseases. The main aim of this paper is to detect the diseases of banana leaf, rose leaf, lemon leaf and beans leaf. This paper presents an algorithm for image segmentation technique which is used for automatic detection and classification of plant leaf diseases. The genetic algorithm is used for image segmentation, which is a crucial component of disease detection in plant leaf disease. At first, to obtain the desired picture region, the leaf image is clipped, and then the smoothing filter is used to smooth the image. The color co-occurrence approach is used for feature extraction. The co-occurrence features are computed after mapping the R, G, and B components of the input picture to the thresholder images. The co-occurrence features for the leaves are extracted, and the associated feature values stored in the feature library are compared with the extracted co-occurrence features. In order to determine a picture's distinctive qualities, a method is used in which both the texture and color of the image are taken into account. First, the classification is carried out using K-Mean Clustering and the Minimal Distance Criterion, demonstrating its effectiveness with an accuracy of 86.54%. With the suggested technique, the detection accuracy is increased to 93.63%. With an accuracy of 95.71 percent, the SVM classifier is used to perform classification in the second phase. Using SVM, the suggested technique increases detection accuracy to 95.71%.

A project has been done in **[7]** by Department of Electronics and Telecommunication, Vishwakarma Institute of Technology, Pune, India, about plant disease detection using image preprocessing and Machine learning. In this project, they used 87000 RGB image of healthy and unhealthy images of plant leaves where they used 25 classes of images. For image preprocessing and feature extraction, they used Gray Level Co-occurrence Matrix or GLCM method. After that, they used random forest classifier for classification or detection. They have split their dataset into 80% training set and 20% validation testing. For finding the accuracy score, they have used K-fold cross validation technique. Their model achieved an accuracy of 93%, which is considered as best case for any machine learning algorithm.

# CHAPTER 3: METHODOLOGY

In this chapter, an overview of different parts of the work is given chronologically. This chapter mainly discusses the theories, techniques, technologies and step by step workflow of the work.

## 3.1 Workflow

In our project, our model will be a CNN model. There are multiple steps of building a CNN model. A workflow diagram displaying the complete process of the method is shown below:

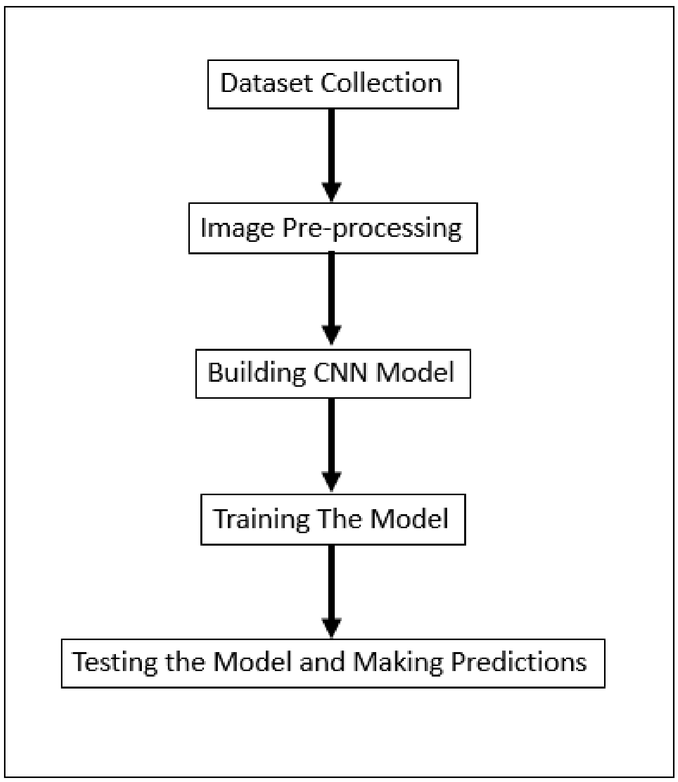


*Figure 1: Block diagram of workflow*

In the diagram, we can see that our first step is collecting a suitable dataset which we can use for training and testing our model. The images from the dataset have to be pre-processed before we can use it for training and testing. After training the model, we can test the model by making predictions on some images and validating the prediction with the actual image and also find the accuracy.

## 3.2 Data Collection

Collecting dataset is our first step and also one of the most important step. Before collecting a dataset, there are some features that has to be explored. Since our project requires an image dataset, we have to look at features like the number of image samples, variations of image, background, image format and resolution.

We have collected our dataset from [8] which contains .jpg type of image. There are ten classes of five different types of plants containing 2273 pictures of both healthy and diseased leaves. All the images in the dataset has similar background. The resolution of all images are also same.

## 3.3 Pre-processing

Before using our images for training, the images have to be pre-processed. Pre-processing is required to increase the quality of input image so that the performance of the model can be improved. We followed the following steps for pre-processing our images:

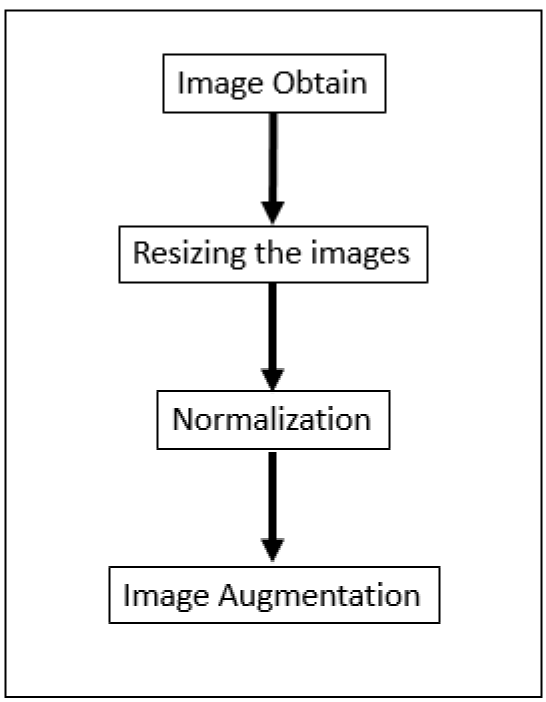


*Figure 2: Block diagram of image pre-processing*

From the above diagram, we can see that after obtaining the image from the dataset, we will be resizing the images. Then we will normalize the images and finally apply image augmentation.

### 3.3.1 Resizing

We started our pre-processing with image rescaling and resizing. Our images from the dataset were 6000x6000 resolution. We resized the image into 256x256 size to ensure that all images stay at a consistent scale.

### 3.3.2 Normalization

After resizing the images, we normalized the images. Normalization keeps all the input pixels on similar distribution. In our model, we performed normalization by dividing the pixel values by 255. By dividing the pixel values, we rescale the values between 0 and 1.

For our model, we performed normalization implicitly in the resizing and rescaling step.

### 3.3.3 Image Augmentation

Our final step of pre-processing stage is image augmentation. Data augmentation is applied to the existing dataset to artificially increase the size of the dataset by performing various transformations like flipping, distorting, rotating, zooming etc. Data augmentation helps to create variation and lowers the chances of overfitting.

In our model, we have randomly flipped the images horizontally and vertically. We have also rotated the images randomly up to 0.2 radians.

## 3.4 Building The Model

Our Machine Learning model is a CNN model. Our model can be divided into two parts, Feature extraction and image classification. We have used CNN for both feature extraction and image classification.

### 3.4.1 Feature Extraction

Feature extraction transforms raw data into numeric values compatible with machine learning algorithms [9]. These numeric values can be processed by keeping the integrity of the information that is found from the original dataset. For our project, we used CNN for feature extraction. There are two main layers of feature extraction in a CNN model. If we take a look at the CNN architecture,


*Figure 3: CNN model architecture of feature extraction*

Here we can see that there are two layers in feature extraction, Convolutional layer and Pooling layer.

Convolutional layer helps to detect various features and pattern which are present in the image by applying convolutional filters to the input image. This filters then learn extracting different visual features like textures.

Pooling layer selects the minimum value within a specific region of the image and down samples it. This helps to retain the relevant information of the image by reducing the relative dimension.

By stacking these two layers, the model can learn how to identify features from the input image.

Our model also has these two layers. We have used two convolution layers and two pooling layers. The convolution layers of our model have 32 filters and each filter has 3x3 kernel size. We have used the activation filter ReLU (Rectified Linear Unit) which is a popular activation function. ReLU uses the formula,

$$f(x) = \max(0, x) \tag{1}$$

Which means that the output can be found only if the input value is positive, otherwise it will be zero. If there is any negative input, ReLU deactivates the neuron by converting it to zero[]. As a result, exponential growth in computation can be prevented. Our two pooling layers capture the most important features with a pool size of 2x2.

### 3.4.2 Image Classification

In a CNN model, the classification layer determines the predicted value on the activation map **[10]**. Once the model has extracted features, it needs to classify those features into learn the features of the data. CNN classification also has two layers. If we look at the classification architecture,

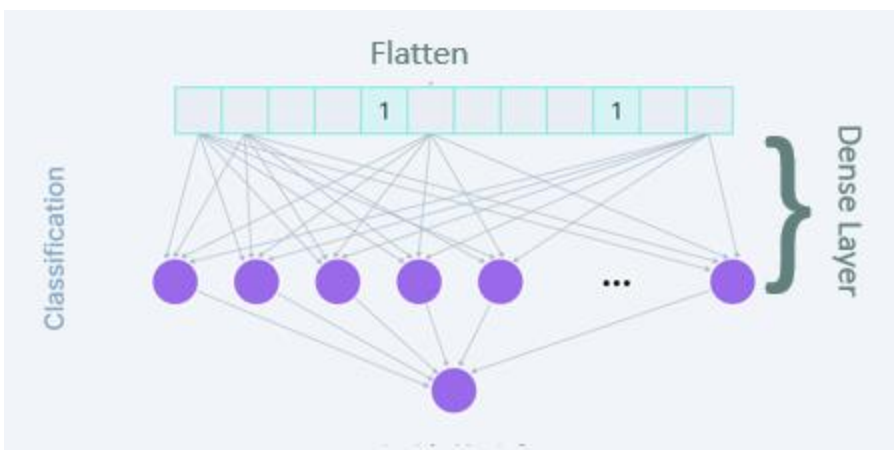


*Figure 4: CNN model architecture of classification*

We can see that there is a flatten layer and a dense layer.

The flatten layer flattens the 2D layers from feature extraction into 1D vector. This layer converts the features into a format which can be processed by the dense layers.

After flattening, all the layers are passed to the dense layer or fully connected layer. Dense layer contains multiple neurons which are fully connected to each other. This layers learns to classify the extracted features into their corresponding classes.

For image classification, we have also used these two layers. The flatten layer flattens the 2D feature maps into a 1D vector before going to the dense layer. Our first dense layer is a fully connected layer consisting of 64 units or neurons. This layer has ReLU activation function which applies linear transformation to the input data. Our second dense layer is the output layer of the model. We have used softmax activation function here. Each output value of this layer represents the probability of input of a particular class.

```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 sequential (Sequential)     (32, 256, 256, 3)         0

 conv2d (Conv2D)             (32, 254, 254, 32)        896

 max_pooling2d (MaxPooling2D  (32, 127, 127, 32)       0
 )

 conv2d_1 (Conv2D)           (32, 125, 125, 32)        9248

 max_pooling2d_1 (MaxPooling  (32, 62, 62, 32)         0
 2D)

 flatten (Flatten)           (32, 123008)              0

 dense (Dense)               (32, 64)                  7872576

 dense_1 (Dense)             (32, 10)                  650

=================================================================
Total params: 7,883,370
Trainable params: 7,883,370
Non-trainable params: 0
_________________________________________________________________
```

*Figure 5: Model summary*

## 3.5 Training The Model

After preprocessing and building the model, our model is ready for training. We have split our dataset into two partitions. The first partition is our training dataset. Which is 80% of the total dataset. And the remaining 20% is the testing dataset. We further split the testing dataset into two more partitions. Which are test and validation. Each partition is 10% of the testing dataset.

We started training our model with the train dataset. The validation dataset is used to evaluate the model's performance. We used the loss function Sparse Categorical Crossentropy. This function calculates the cross-entropy loss between the true labels and the predicted labels. The value increase when the difference increases. We also used Adam optimizing algorithm. Adam can adapt the learning rate of each parameter also maintaining a moving average of squared gradients. As a result, Adam provides stable and fast optimization.

At first, we started training our model with a batch size of 32. We also had an epoch value of 50, which is the number of iterations the model will go through over the entire training dataset. With this values, our did not have a stable accuracy. So, we had to tune the parameters. We tried different batch sizes, epoch values, and learning rate of the optimizer and found the best accuracy with a batch size of 32, epoch 30 and learning rate of 0.001.

## 3.6 Testing The Model and Making Predictions

After completing the training part, we tested our model and made predictions. At first, we evaluated the model on the test dataset and found the loss and accuracy values. For prediction, at first we selected one image sample from test dataset and used the model to make predictions on it. The model was able to make correct predictions. Then we selected 9 images from the test dataset and made predictions. This time, all the predictions were mostly correct. We also displayed the confidence level by taking the maximum predict probability.

# Chapter 4: RESULTS & DISCUSSION

In this chapter, we discuss the results and the overall performance of our model. We will be analyzing our model's performance using some well-known matrices.

Confusion matrix is one of the most common performance measurement used in machine learning. It shows the summary of a model's performance on a test data. This matrix uses the true labels and the predicted labels to show four types of information. These are True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN)

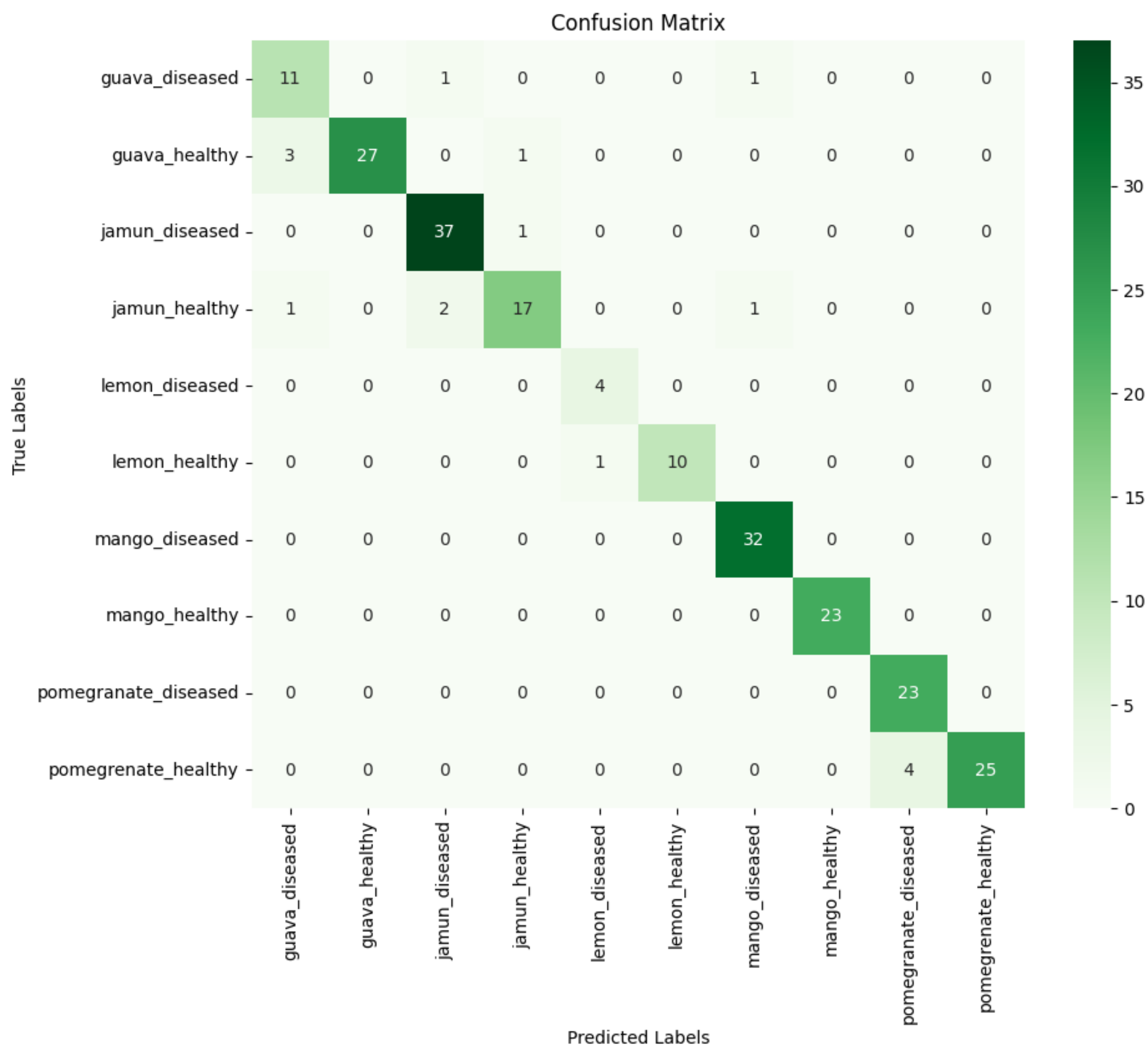


*Figure 6: Confusion matrix*

From the confusion matrix, we can also calculate some important data for each class which we can use to analyze the model. Which are Accuracy, Specificity, Sensitivity, Precision and F1 score.

Accuracy shows the outputs that are actually correct. It is calculated using the formula,

$$\frac{TP+TN}{TP+FN+FP+TN} \tag{2}$$

Specificity is the model's performance on identifying true negative cases, which is calculated using the following formula,

$$\frac{TN}{TN+FP} \tag{3}$$

Sensitivity is the model's performance on identifying true positive cases. Sensitivity is also known as recall. This is calculated using the formula,

$$\frac{TP}{TP+FN} \tag{4}$$

Precision shows how many true positives are predicted correctly. Which means finds how many true positives are actually true positives. The formula which is used to calculate precision is,

$$\frac{TP}{TP+FP} \tag{5}$$

F1 score can be calculated from precision and Recall. It is the harmonic mean of the precision and recall **[11]**, which is calculated by,

$$\frac{2TP}{2TP+FP+FN} \tag{6}$$

We have calculated all these values for each class of our model from the Confusion Matrix, and got the following scores,

| Class Name | Accuracy | Specificity | Sensitivity | Precision | F1 Score |
|---|---|---|---|---|---|
| guava_diseased | 0.973 | 0.981 | 0.846 | 0.733 | 0.785 |
| guava_healthy | 0.822 | 1.00 | 0.870 | 1.00 | 0.931 |
| jamun_diseased | 0.822 | 0.983 | 0.973 | 0.925 | 0.948 |
| jamun_healthy | 0.733 | 0.990 | 0.809 | 0.894 | 0.850 |
| lemon_diseased | 0.995 | 0.995 | 1.00 | 0.800 | 0.888 |
| lemon_healthy | 0.995 | 1.00 | 0.909 | 1.00 | 0.952 |
| mango_diseased | 0.991 | 0.989 | 1.00 | 0.941 | 0.969 |
| mango_healthy | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| pomegranate_diseased | 0.982 | 0.980 | 1.00 | 0.851 | 0.920 |
| pomegranate_healthy | 0.982 | 1.00 | 0.862 | 1.00 | 0.925 |

*Table 1: Model evaluation matrices*

We also have two graphs which can show the performance of our model. The first Graph is the Training and Validation accuracy graph. This graph shows the progression of accuracy of the model over each epoch. The training accuracy is the accuracy of the model on the training dataset and the validation accuracy is the accuracy on the validation dataset which is different from the training dataset.

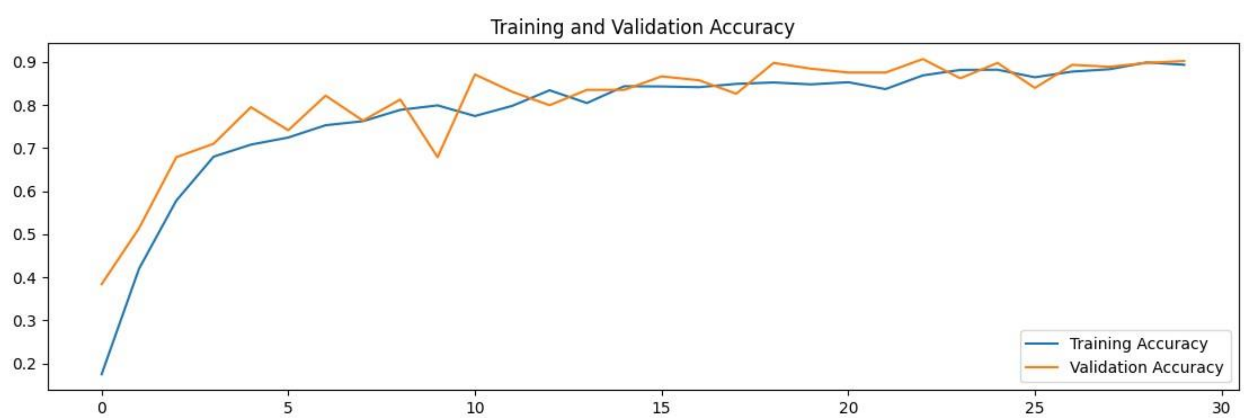


*Figure 7: Training and validation accuracy graph*

From the graph, we can see that the accuracy starts to increase as the model learns to adjust the parameters. We can also see that the training and validation curves are increasing at similar level which shows that the model is learning well.

Our other graph is Training and Validation loss graph. This graph shows the change loss of the model over each epoch. The goal here is to minimize the amount of loss over each epoch.

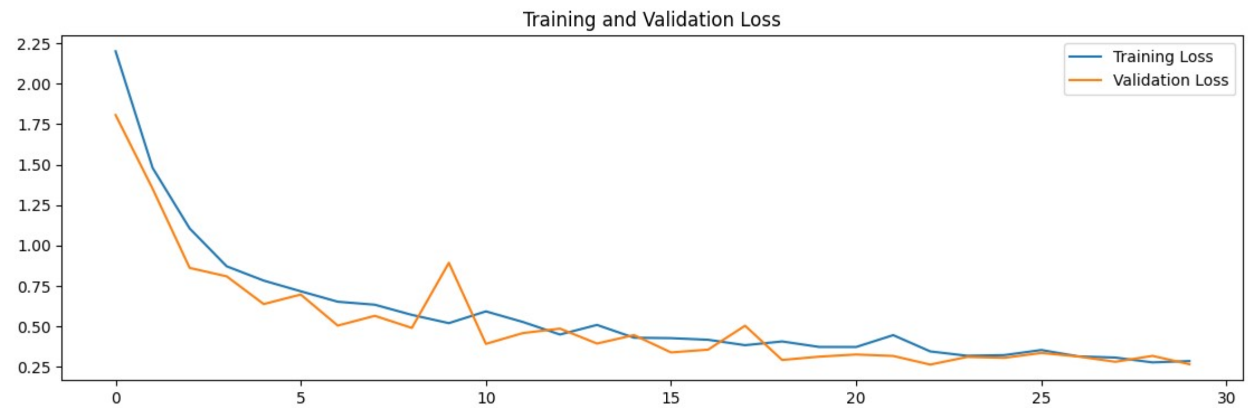


*Figure 8: Training and validation loss graph*

In this graph, we can see that the loss amount starts to decrease with each epoch. The training and validation curves are also on similar position.

# CHAPTER 5: CONCLUSION

In conclusion, we can all agree that leaf disease detection using Convolutional Neural Network is a revolutionary approach in the field of agriculture. It offers a significant solution to the issues of plant health monitoring. In our project, we have successfully built an automated system that can detect leaf diseases in plant leaves by using modern technologies such as machine learning, image processing and data analysis. Since our project only focuses on the plants that are usually cultivated in the rooftop of the urban society. We faced several challenges. The most prominent challenge was to find the proper dataset. But despite the challenges of dataset diversity and labelling accuracy, our CNN model offers several benefits, such as early detection and accurate classification of diseases. This will help the urban people to take timely intervention measures that can revolutionize plant disease management and create a healthier, more resilient cities and develop a better bond between people and nature by incorporating nature into urban settings.

## 5.1 Future Scopes

To increase the system's efficiency, it is crucial to keep improving the algorithms, diversifying the dataset, and adding real-time monitoring. So, in future we intend to accommodate a much larger dataset for our model. Also, we plan to add some features, including human verification, prompt suitable remedies, community interactions and multi-platform availability.

# REFERENCES


[1] “City heat can be reduced by Rooftop Gardens”, Prothomalo. [Accessed: April 8, 2023]

Available at:

https://en.prothomalo.com/environment/research-city-heat-can-be-reduced-by-rooftop-gardens

[2] “Leaf spot” (2022) Wikipedia. Wikimedia Foundation [Accessed: April 8, 2023] Available at:

https://en.wikipedia.org/wiki/Leaf_spot

[3] Horst, R. (2008). Westcott’s Plant Disease Handbook (Seventh Edition.). Springer Netherlands.

[4] Ethem Alpaydin (2020). Introduction to Machine Learning (Fourth Ed.). MIT. pp. xix, 1–3, 13–18. ISBN 978-0262043793.

[5] Leaf Disease Detection Using Convolutional Neural Network, Sneha Patel

https://bvmengineering.ac.in/NAAC/Criteria1/1.3/1.3.4/18CP812_Thesis.pdf

[Published: July 2022]

[6] Singh, V., & Misra, A. (2017). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information Processing in Agriculture*, *4*(1), 41-49. https://doi.org/10.1016/j.inpa.2016.10.005

[7] Kulkarni, Pranesh & Karwande, Atharva & Kolhe, Tejas & Kamble, Soham & Joshi, Akshay & Wyawahare, Medha. (2021). Plant Disease Detection Using Image Processing and Machine Learning.

https://www.researchgate.net/publication/352643083_Plant_Disease_Detection_Using_Image_Processing_and_Machine_Learning

[8] Healthy vs. Diseased Leaf Image Dataset

https://www.kaggle.com/datasets/amandam1/healthy-vs-diseased-leaf-image-dataset

[9] The Role of Feature Extraction in Machine Learning | Snowflake. Snowflake.

https://www.snowflake.com/guides/feature-extraction-machine-learning

[10] *What Is a Convolutional Neural Network? A Beginner's Tutorial for Machine Learning and Deep Learning*. FreeCodeCamp.

https://www.freecodecamp.org/news/convolutional-neural-network-tutorial-for-beginners/

[11] Wikipedia contributors. (2023, April 4). F-score. In *Wikipedia, The Free Encyclopedia*. Retrieved 00:45, June 16, 2023, from https://en.wikipedia.org/w/index.php?title=F-score&oldid=1148225663