*Plant Disease Classification*

Muhammad Khaleduzzaman   
*Department of Computer Science and Engineering*  
*Ahsanullah University of Science and Technology*Dhaka, Bangladesh  
180104102@aust.edu

Md. Tamzidul Islam  
*Department of Computer Science and Engineering*  
*Ahsanullah University of Science and Technology*Dhaka, Bangladesh  
180104123@aust.edu

Tanzela Nasrin Ahana  
*Department of Computer Science and Engineering*  
*Ahsanullah University of Science and Technology*Dhaka, Bangladesh  
180204012@aust.edu

Md. Mubayeer Rahman  
*Department of Computer Science and Engineering*  
*Ahsanullah University of Science and Technology*Dhaka, Bangladesh  
180104144@aust.edu

*Abstract*— Multiple diseases have the potential to drastically impair crop productivity, seriously jeopardizing food security. Therefore, it is critical and vital to accurately detect plant illnesses. Traditional classification systems, such as visible observation and laboratory assessments have many flaws, including the need for a lot of time and perspicacity. Convolutional neural networks (CNNs) have found extensive use in the classification of plant diseases. They illustrate cutting-edge technology in this field and completely or partially resolve the drawbacks of traditional categorization techniques. In this study, we examined the most recent CNN networks that were relevant to the classification of plant diseases. We outlined the DL concepts used to categorize plant diseases. Additionally, we outlined CNN's primary issues and their associated fixes for classifying plant diseases. Then we talked about the way that plant disease categorization would take in the future.

Keywords—plant disease classification; deep learning; machine learning; convolutional neural network.

# Introduction

The development of food crops is significantly impacted by the many plant diseases. Bangladesh is an agricultural country. Modern methods should be used to solve agricultural issues in order to meet the demand for food. The agricultural industries are concentrating on artificial intelligence techniques in this regard. Traditional machine learning (ML) techniques have been applied to a variety of agricultural tasks. Additionally, deep learning (DL) led to important advancements in the study of agriculture. This is because deep learning algorithms have the ability to automatically extract features. Among other agricultural issues, correctly classifying plant diseases is essential to increasing the quantity and quality of agricultural output while minimizing the need for chemical sprayers like fungicides and herbicides. Therefore, advancing agricultural automation is a growing research area in our nation. Due to the similarity in the occurrence of plant diseases, this agricultural task is complex. Numerous studies have been done in this area to better categorize plant diseases. However, visual examination of the leaf color patterns is still the primary method used in conventional field scouting for crop diseases. On the basis of experience and observation of disease symptoms on plant leaves, people can make a skilled and labor-intensive diagnosis of a plant disease. If the farmers can somehow afford a smartphone it will be easier for them to use image processing methods to detect plant diseases.

Agriculture is increasingly using deep learning (DL) techniques, notably those works as convolutional neural networks (CNNs), for tasks including weed detection, agricultural pest categorization, and plant disease diagnosis.

# Background Study

CNN is a type of Supervised Deep Learning. Five main processes make up the CNN process: Convolution, Max Pooling, Flattening, Full Connection.

**Convolution:** The most crucial component of CNN is the convolution layer. Where the filter is applied to the source image or other features of the model is provided by this layer. The size of the filters is typically smaller than the actual one. A feature map is the generated image. The better the filtering process works for us, the higher the number on the featured map, which tells us that we aren't losing many features.

**ReLU Layer:** ReLU layer - Rectified Linear Unit, after Convolution it is an additional step. ReLU is used for increasing the non-linearity. The reason we want to escalate non-linearity for the reason of highly non-linear property of images, if we found linearity we may face problem. To overcome this problem we are more likely to separate linearity.

**Max Pooling:** This pooling operation is also used to determine the maximum value for a feature map. It is then applied to produce a feature map that has been down sampled. We can extract low level features with its assistance. Because of this, this method doesn't care whether the features are close together or are located in different environments.

**Flattening:** The Flattening method means to rearrange Feature Map which is pooled into single column. Flattening covert the data that are currently positioned 2 dimensional array are with mapped pool feature into a single column continuous linear vector for inputting to the next layer.

**Full Connection:** The vector that has been flattened, as mentioned above, is merely a feed-forward neural network technique. Full connection means that the hidden layer is fully connected. The input for fully connected layers actually comes from the convolution or final pooling output.

Our objective is to combine features into additional attributes for improved class prediction. In that case, combining increasingly more attributes enables us to forecast images more accurately.

# Literature Review

Emanuel Cortes used a deep neural network and semi-supervised algorithms were trained to recognize crop species and disease status of 57 distinct classes using a publicly accessible dataset of 86,147 photos of sick and healthy plants. Rs-net was the unlabeled data experiment that performed successfully. It achieved a training phase score of more than 80% in fewer than 5 epochs with a detection rate of 1e-5. [1]

Identification and detection of plant diseases By employing straightforward leaf photos of healthy and sick plants, Konstantinos P. Ferentinos and colleagues constructed CNN models to recognize and diagnose diseased plants. 25 distinct plant species from 58 different classes of [plant, sickness] combinations, including unaffected plants, were included in the 87,848 pictures from an open collection that served as training data for the models. Several model designs were developed; the one with the highest success rate had a success percentage of 99.53%. The model is a useful or early detection tool due to its high success rate.[2]

The process for precisely identifying apple leaf diseases is described in this paper. To detect apple leaf infections, it is necessary to construct a sufficient number of unhealthy images and a deep CNN with a distinctive architecture based on AlexNet. The proposed deep CNN model is designed to identify four prevalent disorders in sick apple leaves using a database of 13,689 images of sick apple leaves. The proposed illness detection model has a 97.6% overall accuracy. The suggested model's parameters were reduced by 51,206,928 when compared to the AlexNet model, and its accuracy increased by 10.83% thanks to the pathological images it produced. According to this study, the disease detection deep learning model may be more accurate and have a faster convergence rate, which would improve controlling the diseases. [3]

# Data Collection Process

Plant Village Dataset is employed. The 54303 healthy and unhealthy leaf images in the Plant Village dataset are categorized into 38 groups based on species and disease.



Figure 1: Images from dataset



Figure 2: Dataset breakup

Our study's dataset consisted of images of the leaves of tomato and potato plants. So, we worked on 2 categories.

Potato plant pictures are divided into three classes: the healthy leaves, early blight, and the late blight. The amount of data used is 1000 for early blight, 152 for healthy leaves and 1000 for late blight.

On the other side, tomato plant pictures are divided into two classes: the healthy leaves and the mosaic virus. The amount of data used is 373 for mosaic virus and 1591 for healthy leaves.

Below we show the figure of our image data:

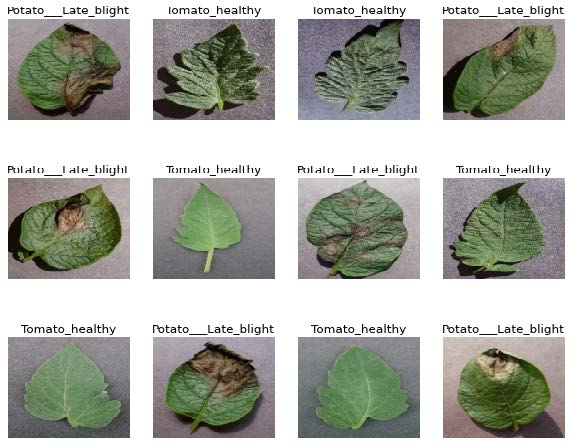


Figure 3: Test input for CNN.

# Methodology

Image augmentation is a necessary component of a strong image classifier. Although datasets may contain hundreds to a few thousand training examples, there may not be enough variability to build a trustworthy model. Resizing the image, rotating it at various angles, and flipping it vertically or horizontally are just a few of the numerous photo enhancing options. The amount of relevant data in a dataset is increased by these additions. The images had different pixels sizes. It would be difficult to train an image of such size since the processing would take too long. So, the images were resized to 256 × 256 pixels so that only the features needed to successfully classify the disease would be present.

Two types of feature extractors were used for this purpose. They are as follows: [4]

## Gabor Filter

The Gabor filter is an application of the Gabor transform, which combines a Gaussian window and a short-term Fourier transformation for analysis in the spatial domain. The texture information of the image is incorporated into the distortion information of content adaptive image steganography. Due to its spatial selectivity and direction, the two-dimensional Gabor filter represents the texture information. Because of its spatial representation, a filter represents texture information.

gλ,θ,ϕ,σ,γ (x,y) = exp() cos(2π)

where, the value of x and y denote the following:

x = acosθ + bsinθ

y = −acosθ + bsinθ

λ – Wavelength of Gabor function cosine factor.

θ – Orientation of Gabor function normal to the parallel stripes.

ϕ – Phase offset of the of Gabor function cosine factor.

σ – Standard deviation sigma of Gaussian factor.

γ – Ellipticity of the Gaussian factor.

A picture containing square

Description automatically generatedBased on different orientation parameters, different types of filters may be generated

Figure 4: Different Orientations for Gabor Filter

## Sobel Filter

The Sobel filter is a popular method to calculate partial derivatives. At each pixel in the image, the gradient approximations given by Gx, and Gy are combined to give the gradient magnitude, using:

G =

The gradient’s direction can be calculated using:

Θ = arc tan

## Deep Learning Models

The DL model such as CNN has been used for the classification.

***Convolutional Neural Network (CNN)***

Convolutional neural networks, a type of artificial neural network, evaluate the inputs of images using a variety of senses and include learnable bases and weights for different characteristics of images that may be used to differentiate one image from another. Using the local spatial coherence of the input images and reducing their weights due to shared features is one benefit of employing a convolutional neural network. Without a doubt, this strategy is effective in terms of complexity and memory.

***Algorithm Steps:***

**Step 1:** Dividing our dataset within train and test data.

We took training data size 0.85 and took test dataset size 0.3.

**Step 2:** Creation of Model Our model input layer made up with 3 classes and activation function named as ‘ReLU’. Hidden layers consist with activation function. Our resultant output layer with 1 node with where activation function named ‘sigmoid’.

**Step 3:** Train our dataset

Our model trained with images where epochs taken as 50 and batch sizes taken as 64.

**Step 4:** Result of CNN model

Using the CNN model, we get our accuracy 98.9%.



Figure 5: Flow chart of CNN

|  |  |  |
| --- | --- | --- |
| **Layers** | **Output Shapes** | **Parameter** |
| Conv2d-1 | [-1, 32, 224, 224] | 896 |
| ReLU-2 | [-1, 32, 224, 224] | 0 |
| BatchNorm2d-3 | [-1, 32, 224, 224] | 64 |
| Conv2d-4 | [-1, 32, 224, 224] | 9,248 |
| ReLU-5 | [-1, 32, 224, 224] | 0 |
| BatchNorm2d-6 | [-1, 32, 224, 224] | 64 |
| MaxPool2d-7 | [-1, 32, 112, 112] | 0 |
| Conv2d-8 | [-1, 64, 112, 112] | 18,496 |
| ReLU-9 | [-1, 64, 112, 112] | 0 |
| BatchNorm2d-10 | [-1, 64, 112, 112] | 128 |
| conv2d\_11 | [-1, 64, 112, 112] | 36928 |
| ReLU-12 | [-1, 64, 112, 112] | 0 |
| BatchNorm2d-13 | [-1, 64, 112, 112] | 128 |
| MaxPool2d-14 | [-1, 64, 56, 56] | 0 |
| Conv2d-15 | [-1, 128, 56, 56] | 73,856 |
| ReLU-16 | [-1, 128, 56, 56] | 0 |
| BatchNorm2d-17 | [-1, 128, 56, 56] | 256 |
| Conv2d-18 | [-1, 128, 56, 56] | 147,584 |
| ReLU-19 | [-1, 128, 56, 56] | 0 |
| BatchNorm2d-20 | [-1, 128, 56, 56] | 256 |
| MaxPool2d-21 | [-1, 128, 28, 28] | 0 |
| Conv2d-22 | [-1, 256, 28, 28] | 295,168 |
| ReLU-23 | [-1, 256, 28, 28] | 0 |
| BatchNorm2d-24 | [-1, 256, 28, 28] | 512 |
| Conv2d-25 | [-1, 256, 28, 28] | 590,080 |
| ReLU-26 | [-1, 256, 28, 28] | 0 |
| BatchNorm2d-27 | [-1, 256, 28, 28] | 512 |
| MaxPool2d-28 | [-1, 256, 14, 14] | 0 |
| Dropout-29 | [-1, 50176] | 0 |
| Linear-30 | [-1, 1024] | 51,381,248 |
| ReLU-31 | [-1, 1024] | 0 |
| Dropout-32 | [-1, 1024] | 0 |
| Linear-33 | [-1, 39] | 39,975 |
| Total parameters: 52,595,399 | | |
| Trainable parameters: 52,595,399 | | |
| Non-trainable parameters: 0 | | |

Table. 1. Convolutional Neural Network Model.

# Result Analysis

CNN model works on the Plant Village dataset and provides us the accuracy. The table below shows the accuracy and loss of our CNN model.

## Figures and Tables

#### Positioning Figures and Tables:

|  |  |  |
| --- | --- | --- |
| Models | Test accuracy | Test Loss |
| CNN | 98.9% | 3.91% |

Table. 2. Test Accuracy and Loss for CNN model.

The graph shown below is for CNN model (a) accuracy graph and (b) loss graph when the train: test is 85:15. The blue line is for training dataset and the orange line is for testing dataset. We see that our model work fine since the accuracy starts increasing gradually and the loss decreases in a stable manner.

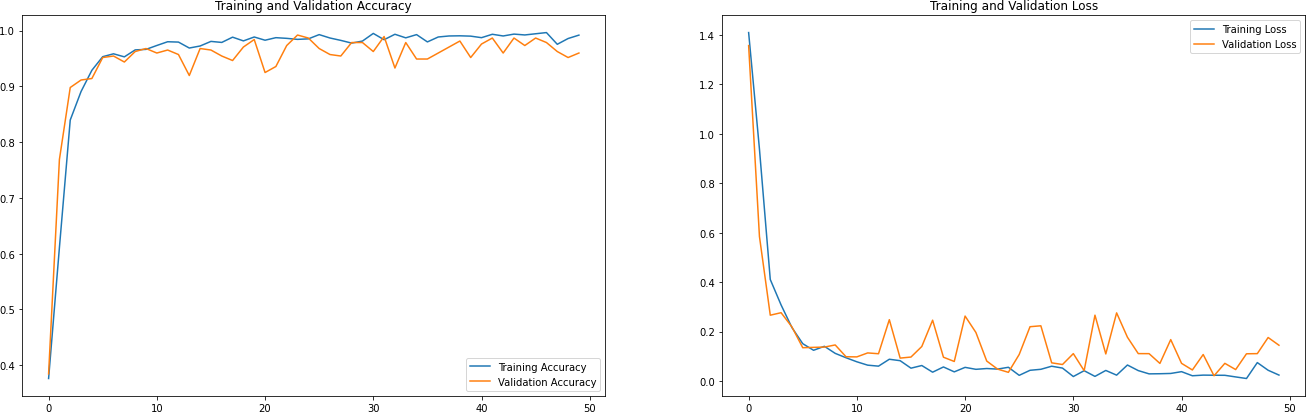


Figure 6: Training and validation from 85:15 data dividing (a) accuracy (b) loss of CNN

The table 2 shows the accuracy values of CNN model at each epochs for train-test ratios at 85:15. On the 40th epoch the accuracy comes 99% and the loss was 2.7%. On the final epoch the accuracy was 99% on training phase and 98% on test phase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Epoch** | **Data Training** | | **Data Testing** | |
| Acc | Loss | Val ACC | Val Loss |
| 40 | 0.9903 | 0.0273 | 0.9651 | 0.1164 |
| 41 | 0.9939 | 0.0202 | 0.9839 | 0.0845 |
| 42 | 0.9851 | 0.0421 | 0.9516 | 0.1907 |
| 48 | 0.9912 | 0.0264 | 0.9919 | 0.0504 |
| 49 | 0.9912 | 0.0238 | 0.9651 | 0.1767 |
| 50 | 0.9912 | 0.0263 | 0.9839 | 0.0698 |

Table. 3. Result from fit model 85:15 (CNN)

Here are some images of our final findings, which highlight the potency and effectiveness of our study.

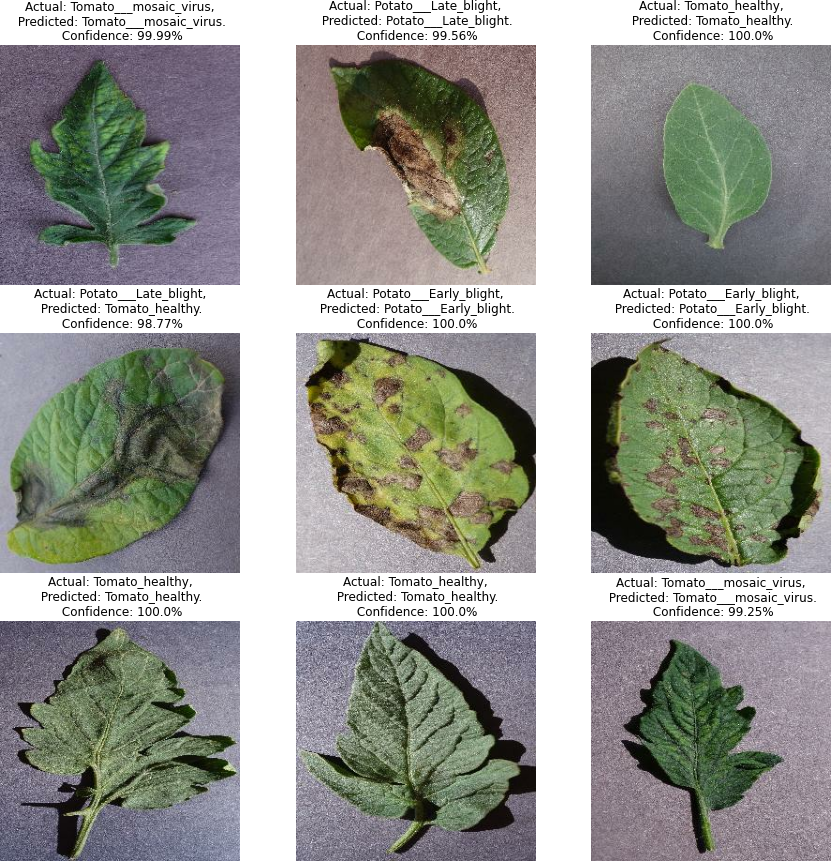


Figure 6: Test output for CNN

# Conclusion

Since a large portion of the population in Bangladesh depends on agriculture, it is crucial to find and identify the leaf diseases that cause losses because agriculture is so important to the country's economic development. The CNN project, which is based on deep learning, is used to create a system for identifying, detecting, and recognizing various plant diseases. The Plant Village dataset is used to train the neural network. A convolutional neural network that had been trained to identify and recognize plant leaf disease was able to correctly classify and predict diseases for nearly all images with only a few anomalies, achieving an accuracy of 98.9%.

# References

1. Cortes, Emanuel. ”Plant disease classification using convolutional networks and generative adversarial networks.” (2017).
2. Wallelign, Serawork, Mihai Polceanu, and Cedric Buche. ”Soybean plant disease identification using convolutional neural network.” The thirty-first international flairs conference. 2018.
3. Mohanty, Sharada P., David P. Hughes, and Marcel Salath´e. ”Using deep learning for image-based plant disease detection.” Frontiers in plantscience 7 (2016): 1419.
4. S Ramesh,D.Vydeki“Rice Blast Disease Detection and Classification Using Machine Learning Algorithm.”,2021.