Prediction of Plant Disease using Convolutional Neural Network
(Bioimage analysis and extended phenotyping)

Lecturer:
Dr. Christian Kappel

Author:
Ahmed Sameed
Matriculation # 824974

Table of Contents

Abstract:

This report deals with the prediction of diseases in plants based on their leaf images. To accomplish this task, a convolutional neural network (CNN) is trained on three types of plant leaves namely tomato, potato, and corn. The dataset has corn leaves with four categories (blight, common rust, gray spot, and healthy); potato leaves with three categories (early blight, late blight, and healthy) and tomato leaves with five categories (bacterial spot, black mold, gray spot, late blight, and healthy). The training and evaluation of CNN involved four steps namely, Dataset preprocessing, Model architecture, Model training and Model evaluation. Dataset preprocessing consists of resizing images from 256x256 pixels to 128x128 pixels. The model architecture includes a sequential convolutional neural network having convolutional, pooling, dropout, and fully connected layers. Adam is used as a default optimizer for model optimization and categorical cross-entropy loss is used as a loss function as it is suitable for multiclass classification. The dataset is divided into train and test sets. 80% of data is used for training and 20 % is used for testing. For model training, a default batch size of 64 is used. After 50 epochs, the model attained train accuracy of 92.87% and a test accuracy of 83.99%. To evaluate the CNN, F1 score, and classification report metrics are used to compare true and predicted labels, corroborating the model's effectiveness. This study underlines the effectiveness of using CNNs in agriculture-based image classification.

1. Introduction:

Plant diseases are a severe threat to food security as they act as major hinderance in total agricultural yield. The United Nations Food and Agriculture Organization reports that certain plant diseases and pests cause a 40% annual loss in crop yield (Nations, 2024). Since agricultural products are important to a nation's economy, any decline in these goods directly affects the nation's economy.

Due to a lack of access to adequate infrastructure, increasing agricultural productivity is a challenging task worldwide. As a result, the proper identification and treatment of plant diseases continues to be a challenging problem. Recent developments in computer vision and image processing come across as being a major step toward the identification of plant diseases, ensuring healthy crop production, and improving the overall yield and quality of agricultural products.

Image classification acts as a steppingstone in the field of computer vision, where the goal is to correctly assign a label to an image by classifying it into one of a predefined set of categories. Now with CNN in play, the accuracy and efficiency of image classification has significantly improved. This report deals with the development and evaluation of CNN model for image classification task to detect diseases in plant leaves. The goal here is to distinguish between the healthy leaves from the diseased ones and correctly predict which kind of disease they have. For this purpose, images of tomato, corn and potato leaves, both healthy and diseased, are considered and a convolutional neural network has been trained to make predictions about their health conditions.

2. Literature Review:

Initially, the task of identifying the diseases in plants was based solely upon human judgment by gathering different leaf samples and studying their molecular pattern using molecular testing or microscopy. The issue with such visual assessment techniques was that it was neither prompt nor precise and could lead to false positives due to mistakes, prejudice, and optical illusions (Manju Bagga, 2024).

A study done by (Santosh Reddy, 2015), recommended the development of an application-based interface which would assist farmers in detecting the diseases in crops, and would suggest remedies in real time, so that the farmers would be able to make informed decisions in time. This would help farmers increase their annual yield by minimizing their losses due to different diseases.

A study done by Singh & Mishra (Misra, 2016) used image segmentation technique to detect and classify diseases in plants. In their model, they employed image segmentation to extract meaningful features from plant leaf images, while neural networks/fuzzy logic was used for disease prediction. Their model achieved an accuracy of 97.6%. They also suggested certain classification techniques which could be deployed in future.

In 2019, Saleem et al. used deep learning models for identification and categorization of plant diseases. They did an in-depth analysis of how different deep learning models perform in context of image recognition and disease detection. Their research highlighted that using deep learning in agricultural practices can lead to improvements in annual crop production (Muhammad Hammad Saleem, 2019).

In conclusion, the inclusion of image processing and deep learning techniques in agriculture, especially for disease detection in plants, has improved the overall production and has influenced the groundbreaking advancements in agriculture.

3. Methodology:

3.1. Dataset description:

The dataset is collected from three different repositories (Rizwan, n.d.) available on an online data science community Kaggle.com. The dataset comprises of leaf images of three plant types namely, tomato, potato, and corn. Each type then has multiple subcategories representing types of diseases and healthy leaves. Tomato has 5 subcategories: bacterial spot, black mold, gray spot, late blight, and healthy leaves. Potato has 3 subcategories: early blight, late blight, and healthy leaves. Corn has 4 subcategories: blight, common rust, gray spot, and healthy leaves. An overview of the image frequency for each subcategory is illustrated in Figure 1.

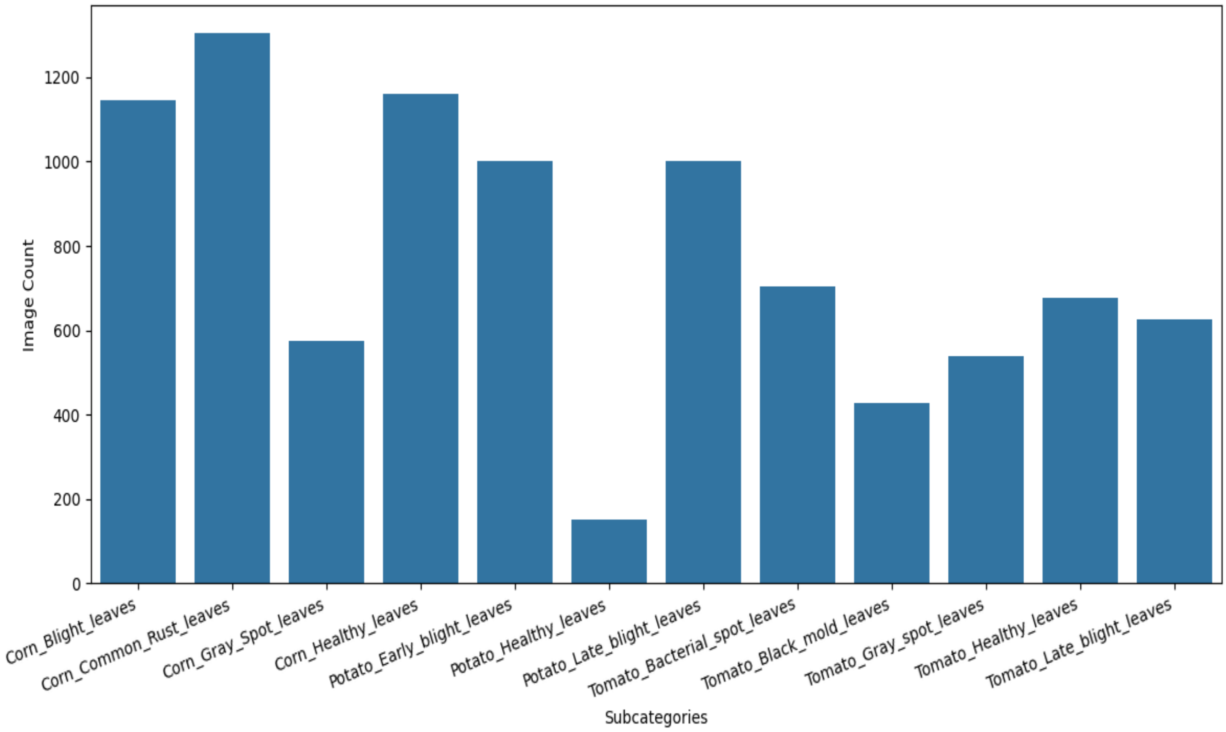

Figure 1: Frequency of images in Subcategories

3.2. Data Preprocessing:

To prepare the dataset for training convolutional neural network, each image is resized to a uniform dimension 128x128 pixels and RGB color channel. The aim for resizing the images is to ensure uniformity in the input and reduction in computational complexity. All the images that don't match the expected shape (128, 128, 3) are discarded. Images are then converted into numpy arrays, and their pixel values are further normalized between [0,1] by dividing with 255. The goal here is to ensure fast convergence while training CNN. Label encoding is employed to encode categorical labels as integers.

Next, the dataset is split into training and testing sets. 80% of data is used in model training while 20% of data is kept aside to evaluate model's performance on unseen data. To ensure the uniformity of input images, the training data is reshaped. The categorical labels, which were previously encoded as integers are converted to one-hot encoded vectors. One-hot encoding provides matrix like representation of class labels, ensuring that the model can learn and distinguish between different classes. An overview of train/test split can be seen in Figure 2.

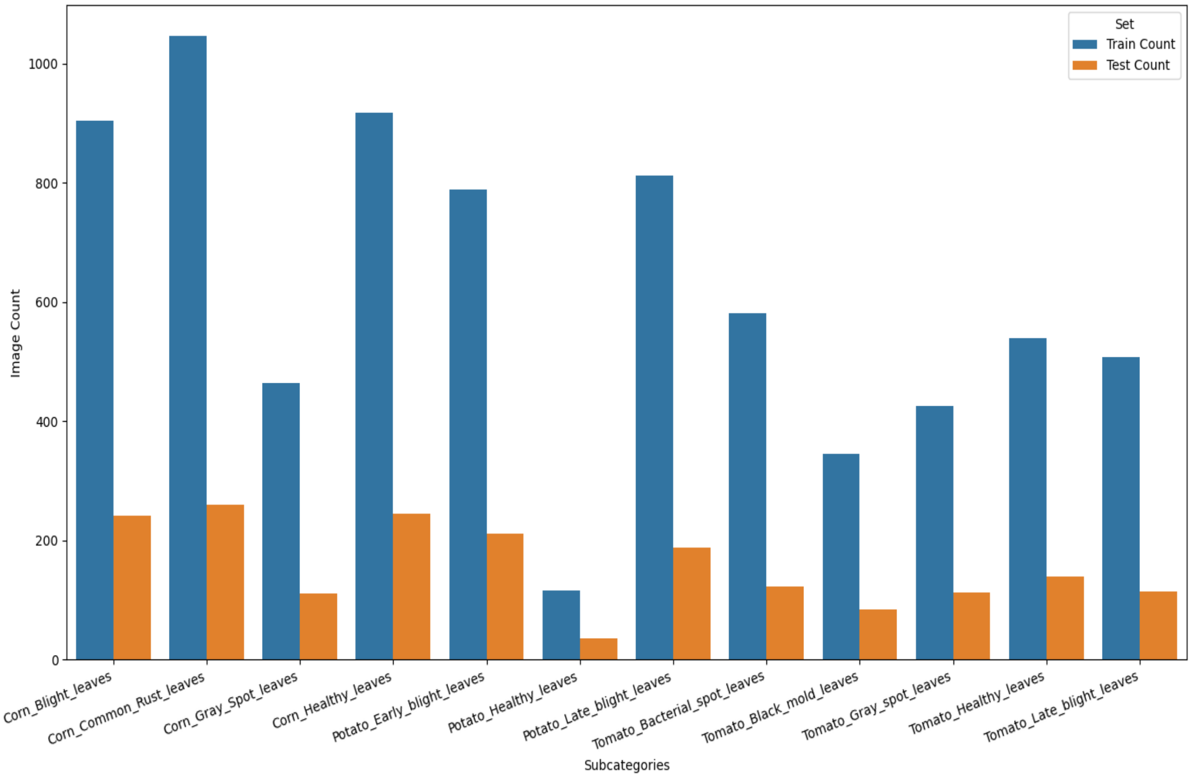

Figure 2: Frequency of Images in Training and Testing Sets by Subcategory

3.3. Model Architecture:

Convolutional Neural Network (CNN) belongs to family of deep learning models that were originated to deal with processing and interpretation of grid like-input, for example images. CNNs has the tendency to identify and process spatial features from raw pixel data and make predictions based on these learned hierarchical features. For this project, keras library is used to initialize CNN for image classification.

In this project, CNN is initialized sequentially with an input layer which takes input data (images in current scenario). The first layer in the architecture is a 2-dimensional (2D) convolutional layer with 32 filters. This layer extracts features from the previous layer using filters with kernel size of 3x3 and generate feature maps. Every feature map captures a specific pattern from the input image. The rectified linear unit (ReLU) is used as an activation function to ensure non-linearity which will help the network to learn complex patterns and to avoid the problem of vanishing gradients.

Following the convolutional layer is a 2D max pooling layer with a pool size of 2x2. This layer reduces the dimensionality of feature maps and filters the meaningful features from input through down-sampling. A batch normalization layer is added next to the pooling layer to normalize the output from the pooling layer.

To further extract features from input images, three additional 2D convolutional layers with 32, 64 and 128 filters (each with kernel size of 3x3 and ReLU activivation) are added, respectively. Following each convolutional layer, there is a 2D max pooling layer (with pool size of 2x2) and a batch normalization layer.

Next up, the feature map is transformed from two dimensions to one dimension via “Flatten” layer. This transformation is done so that data can be sent to the fully connected layer.

The fully connected layer connects the feature map elements to the output neurons. The model has 3 fully connected layers (dense layers). The first layer has 512 neurons and a ReLU activation function. This layer transforms learned features to a high-dimensional representation and helps the model to learn non-linear relationships in data. A dropout layer is added next to the first dense layer with a dropout rate of '0.5'. The role of dropout layer is to prevent overfitting by randomly selecting neurons and setting them to zero at a frequency of dropout rate during model's training. The second dense layer consists of 128 neurons and is followed by a dropout layer with a dropout rate of 0.2.

The third dense layer has 12 neurons equaling the total number of plant categories. Softmax activation function is deployed in this layer which converts the output scores from previous layer to probability values, indicating the likelihood of each class.

For optimization, Adam is used as an optimizer with a learning rate of '0.0001'. The loss function used is categorical cross entropy which is suitable for multiclass classification problem. Two callbacks 'ReduceLROnPlateau' and 'EarlyStopping' are used in the model to monitor validation loss of model. ReduceLROnPlateau reduces the learning rate of model by a factor of '0.1' if the validation loss doesn't improve after 5 epochs. EarlyStopping prevents overfitting by stopping the training process, if validation loss doesn't improve after 10 epochs.

3.4. Model Training:

In model training, an epoch determines that neural network is being trained with all the available training data in one cycle. One cycle consists of a forward pass (prediction) and backward pass (error correction). The batch size determines the number of samples (images) that will be passed from training data to neural network in one iteration.

For this project, the epoch is set to 50. Batch size is set to 64. The training data is split into training and validation sets to observe model's performance during training.

3.5. Results:

To have insight into model's performance, training/validation accuracy and loss were plotted as shown in Fig. 3.

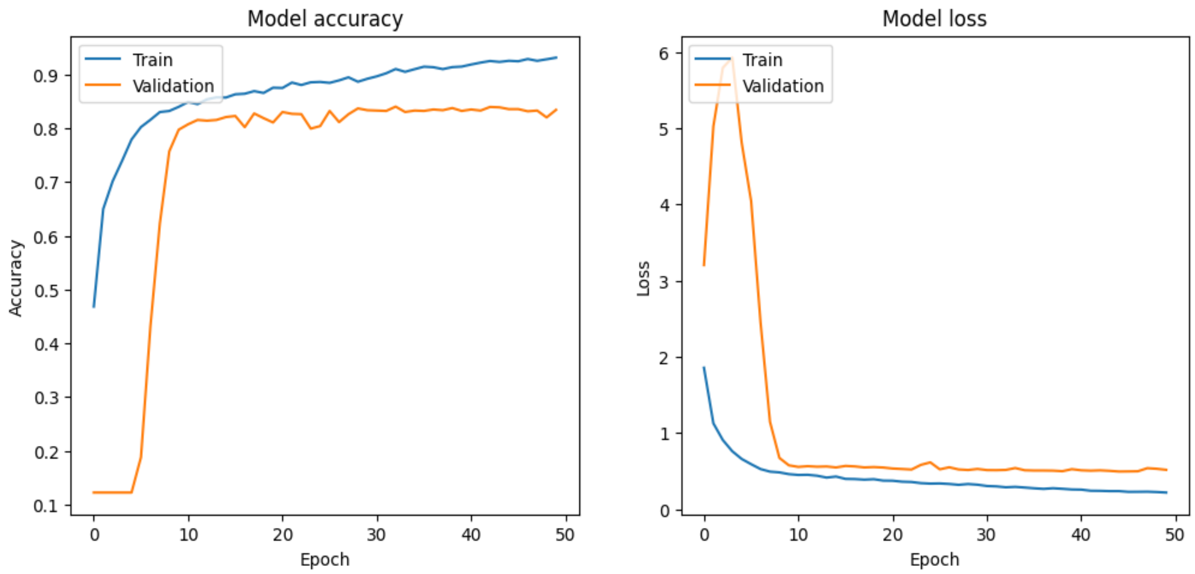

Figure 3: Loss and accuracy progress with epochs.

With the learning rate of 0.0001, training accuracy of model started at 34.6% and with rapid improvement, reached 77.75% with decrease in training loss from 2.39 to 0.65 after 5 epochs. However, the validation accuracy remained static at 12.21% while there is increase in validation loss after 5 epochs, indicating that the model is struggling to generalize initially. After the 5th epoch, the learning rate was reduced by a factor of 0.1.

From 6th to 10th epoch, validation accuracy improved from 18.7% to 79.7% with decrease in validation loss from 4.04 to 0.57, indicating that the model is generalizing well to the validation set. After 50 epochs, the model attained a training accuracy of 92.8% with a training loss of 0.2190. The corresponding validation metrics are 84.3% accuracy and 0.4642 loss.

After the training, the model is evaluated on test data. The model attained test accuracy of 83.9% and loss of 0.4652, indicating that the model is generalizing well to the unseen data.

3.6. Evaluation metrics:

To evaluate model's performance, two evaluation metrics are utilized namely, F1 score and classification report.

The F1 score is the harmonic mean of true positive rate and positive predictive value, ranging between 0 and 1. A higher F1 score (closer to 1) indicates more accurate predictions from the classifier. For our model, the F1 score is 0.8391 on test data, indicating that the CNN model is highly effective in classifying the plant leaf conditions.

The classification report is a tool to evaluate the performance of the classifier. It displays various metrics such as precision, recall, and F1 score for each class The precision is the ratio of true positives to the total number of predicted positives. It measures the accuracy of the positive predictions made by the model. The recall is the ratio of true positives to the total number of actual positives. It measures the ability of the model to correctly identify positive samples. Support is the number of samples that each metric was calculated on (Powers, 2007). The classification report for our model is given below.

Categories	Precision	Recall	F1 score	Support
Corn_Blight_leaves	0.84	0.76	0.80	242
Corn_Common_Rust_leaves	0.92	0.91	0.92	260
Corn_Gray_Spot_leaves	0.63	0.77	0.69	110
Corn_Healthy_leaves	0.96	0.97	0.97	244
Potato_Early_blight_leaves	0.94	0.94	0.94	211
Potato_Healthy_leaves	1	0.80	0.89	36
Potato_Late_blight_leaves	0.90	0.96	0.93	188
Tomato_Bacterial_spot_leaves	0.63	0.73	0.68	123
Tomato_Black_mold_leaves	0.67	0.63	0.65	83
Tomato_Gray_spot_leaves	0.69	0.52	0.59	112
Tomato_Healthy_leaves	0.9375	0.97	0.95	139
Tomato_Late_blight_leaves	0.61	0.61	0.61	114

Table 1: Classification report of the trained model

The model achieved the highest f1 score with corn healthy leaves (0.97) and potato early blight leaves (0.93), indicating strong predictive power for these categories. Conversely, it performs less effectively on tomato gray spot leaves (0.59) and tomato late blight leaves (0.61). Corn and Potato diseases are generally well-classified, with precision and recall mostly exceeding 0.8. However, Tomato diseases present a challenge, with lower precision and recall values, particularly for black mold and late blight leaves.

4. Conclusion:

In this project, a Convolutional Neural Network (CNN) is trained and evaluated on labeled images of three plant types: Corn, Potato, and Tomato, for classification of their leaf diseases. The primary goal is to use the deep learning model in early identification of conditions which affect these plants, ultimately aiding in early detection and reduction in annual loss of crop production.

During preprocessing, all images are resized to 128 x 128 pixels to have a uniform input for model training. After the training, the model is able to reach a train accuracy of 92.87% and a test accuracy of 83.99%, which is quite significant. The model is then evaluated using different classification performance metrices namely, classification report and F1 score.

The F1 score of 0.8391 suggests that model's performance is remarkable in distinguishing between the different leaf conditions.

In conclusion, this project highlights the implementation and effectiveness of using CNN to improve crop quality as well as its production. The model's classification report and high F1 score affirms its usage as a valuable tool for agricultural disease management.

References

Guo, S. (2023). Leaf Disease Detection by Convolutional Neural Network (CNN). *Highlights in Science Engineering and Technology, 72*, (pp. 1141-1146).

Manju Bagga, S. G. (2024). Image-based detection and classification of plant diseases using deep learning: State-of-the-art review. *Urban Agriculture & Regional Food Systems, Volume 9*.

Misra, V. S. (2016). Detection of plant leaf diseases using image segmentation and soft computing techniques. *Information processing in Agriculture*, 41-49.

Muhammad Hammad Saleem, J. P. (2019). Plant Disease Detection and Classification by Deep Learning. *Plants*, 468.

Nahm, F. S. (2021). Receiver operating characteristic curve: overview and practical use for clinicians. *Korean Journal of Anesthesiology*, 25-36.

Nations, F. a. (2024). *Plant Production and Protection*. Retrieved from Food and Agriculture Organization of the United Nations: https://www.fao.org/plant-production-protection/about/en#:~:text=Sustainable%20plant%20pest%20and%20disease%20management&text=Every%20year%2C%20up%20to%2040%20percent%20of%20crops%20are%20lost,to%20plant%20pests%20and%20diseases.

Powers, D. M. (2007). *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation.* Adelaide : School of Informatics and Engineering, Flinders University.

Rizwan, F. G. (n.d.). *plant leaf diseases*. Retrieved from Kaggle.com: https://www.kaggle.com/datasets/krishd123/potato-leaf-disease-gallery; https://www.kaggle.com/datasets/farukalam/tomato-leaf-diseases-detection-computer-vision; https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset

Santosh Reddy, A. P. (2015). A survey on crop disease detection and prevention using android application. *International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 4*, 621-626.