Rose Leaf Disease Detection

Tasnia Sharmin
Master's in Data science
University of New Haven
tshar8@ung.newhaven.edu

Mounika Davuluri
Master's in Data Science
University of New Haven

Abstract

Rose cultivation is a significant component of horticulture, admired for its beauty and commercial value. However, rose plants are susceptible to various diseases, which can severely impact their health and marketability. Accurate and timely detection of these diseases is essential for effective management and maintenance of rose plantations.

In this study, we compared the performance of two popular convolutional neural network (CNN) architectures, ResNet50 and VGG16, for the detection of rose leaf diseases. Our investigation included an analysis of training time, convergence behavior, accuracy, and precision-recall-F1 scores for both models.Our results indicate that ResNet50 demonstrated a shorter training duration but VGG16 exhibited superior accuracy, suggesting better generalization capability.

Overall, our study provides valuable insights into the performance of CNN architectures for rose leaf disease detection. The findings can aid researchers and practitioners in selecting suitable models for disease detection tasks, considering factors such as training time and performance metrics. This study contributes to the advancement of plant disease detection technology, facilitating more effective management practices in horticulture.

Introduction

A sizable fraction of the workforce is employed in agriculture, which is the backbone of Bangladesh's economy. But many farmers still use conventional farming practices and have never received a formal education. This environment has begun to change with the introduction of computer science, which provides novel answers to persistent issues. Plant diseases are a major problem for farmers since they can negatively affect crop productivity and quality.

Assessing the signs and visual patterns that signify a plant's illness is a crucial part to identifying plant diseases. Farmers typically depend on agricultural specialists who perform tests and visually inspect plants to detect diseases. Unfortunately, a lack of specialists impedes this procedure, which causes delayed diagnosis and additional crop disease transmission. Many farmers in nations like Bangladesh lack the expertise necessary to identify illness symptoms early on or know when to consult professionals. The issue is made worse by this ignorance, which permits illnesses to spread and harm more plants before appropriate measures are taken. Prompt identification and action are essential to reducing these losses. Farmers may minimize damage and act quickly by employing modern tools to detect diseases early on in the plant health monitoring process.

The health and vitality of plants, including roses, are essential for agricultural productivity, environmental sustainability, and ecosystem health. The popularity of roses in home gardening, landscaping, and commercial cultivation makes the classification and diagnosis of diseases in rose plants of enormous technical and economic relevance. Roses are widely regarded around the world as the "King of flowers" due to their exquisite beauty. But they can get infected with several diseases that can harm their worth as a commodity, as well as their health and looks. Researchers are examining techniques based on texture, color, and form analysis to automate disease identification in roses. To maximize revenues and minimize expenses, the objective is to enhance output and quality. Infections by fungi, bacteria, and viruses are only a few of the diseases that can harm roses' leaves. Recently, there has been a lot of talk in the rose-growing community about rose rosette disease [1].

Early detection and accurate diagnosis of diseases in plants are crucial for implementing timely interventions and preventing the spread of infections. Traditional methods of disease detection often rely on visual inspection by human experts, which can be time-consuming, labor-intensive, and subjective. In recent years, there has been a growing interest in leveraging machine learning and deep learning techniques for automated and efficient plant disease diagnosis.

Convolutional Neural Networks (CNNs) have emerged as powerful tools for image-based disease identification due to their ability to automatically learn discriminative features from raw pixel data. CNNs have demonstrated remarkable success in various computer vision tasks, including image classification, object detection, and segmentation. In the domain of plant pathology, CNNs offer a promising approach for accurate and rapid disease detection by analyzing digital images of plant leaves. By learning to distinguish between healthy and diseased leaves based on visual features, CNNs can assist farmers, agronomists, and researchers in monitoring plant health, detecting diseases early, and making informed management decisions.

In this study, we focus on the detection of diseases in rose leaves using two widely used CNN architectures: VGG16 and ResNet50. We aim to conduct a comprehensive comparative analysis of these models to evaluate their performance, identify their strengths and weaknesses, and provide insights into their suitability for rose leaf disease detection. The study is motivated by the lack of extensive comparative evaluations of deep learning models for rose leaf disease detection and the growing need for accurate and efficient automated diagnostic tools in agriculture.

Related Work

In recent years, there has been a noticeable increase in interest in the use of deep learning and machine learning approaches for the diagnosis of plant diseases. A variety of methods have been investigated by researchers, such as transfer learning techniques, shallow and deep neural networks, and conventional machine learning algorithms. CNNs can learn hierarchical representations directly from picture data, which has led to their emergence as a dominating paradigm for image-based plant disease diagnosis. Not only deep learning models, researcher are trying to detect using traditional machine learning methods also. One study presents a machine learning-based method for identifying brown spot, bacterial leaf blight, and leaf smut, three prevalent diseases of rice plants. Clear pictures of the impacted rice leaves on a white background are fed into the system. After preprocessing, a number of machine learning techniques, such as K-Nearest Neighbor (KNN), Decision Tree (J48), Naive Bayes, and Logistic Regression, are used to train the dataset [2]. Another study developed Convolutional Neural Network (CNN) models with the goal of categorizing diseases on rose leaves. Up to 15 distinct diseases will be identified depending on their symptoms. Support Vector Machine (SVM)-based hybrid deep learning methods are used to improve classification performance. In order to integrate the output from a fully connected layer, the study investigates early and late fusion algorithms using the VGG16 architecture.The models with early fusion perform better than those with late fusion or VGG16 alone, according to the results [3]. For quick and precise rose disease identification, the study uses a MobileNet model in conjunction with transfer learning. The study makes use of enhanced picture data because it is not readily available. 400 photographs are utilized for the model's testing and 1600 images are used for training. ROC curve analysis, accuracy, and F1 score are examples of evaluation measures. The accuracy of 95.63% is achieved by employing MobileNet with transfer learning, which performs better than the method without transfer learning [4]. Furthermore, one study had done which suggests a two-stage technique called TSDDP for rose pest detection in order to increase the efficacy of identifying illnesses and insect pests in roses, which have a major influence on ornamental value and yield. In order to improve feature extraction and fusion in the YOLOv3 model for improved detection in complicated backgrounds, the method makes use of an optimized Inception module. Additionally, it uses an Anchor box with Faster R-CNN optimized by K-means clustering to enhance target identification of insect pests and rose diseases. In natural settings, the TSDDP approach outperforms YOLOv3 and Faster R-CNN in terms of detection and placement accuracy. The average accuracy of pest and disease detection is 82.26%, which improves the identification and localization of small-scale illnesses and insect pests in roses while decreasing false positives resulting from complicated background conditions [5].

While many studies have investigated the application of CNNs to plant disease detection, there is a paucity of research focusing specifically on rose leaf diseases. Moreover, few studies have conducted comprehensive comparative evaluations of different CNN architectures for this task. This highlights the need for rigorous experimental studies that compare the performance of various deep learning models on rose leaf disease datasets, considering various factors such as accuracy, computational efficiency, and robustness to variations in image quality and disease severity.

Methodology

In this study we are trying to do comparative study on rose plant disease using VGG16 and Rosenet50.

Dataset:

Here we used the Kaggle dataset "https://www.kaggle.com/datasets/warcoder/rose-leaves-disease-detection" to detect rose leaf diseases. The dataset is based on rose leave of Bangladesh. It consists of pictures of rose leaves divided into three groups: fresh leaves, Black Spot, and Downy mildew. The purpose of the dataset is to aid in the detection and classification of frequent illnesses that impact rose plants, an essential responsibility for preserving the health and vitality of both agricultural crops and rose gardens.

Classes:

Disease-Free Leaves: This class shows the images of perfectly healthy rose leaves that show no symptoms of illness or infection. These photos are crucial for training the model to correctly identify healthy foliage since they provide a baseline for comparison.

Black Spot: The photos in this class show rose leaves that have been impacted by Black Spot, a common fungal disease that manifests as black patches on the leaf surface. If black spot is not treated, it can cause defoliation and impair the general health of rose plants.

Downy Mildew: This class includes images of rose leaves exhibiting symptoms of Downy Mildew, another fungal disease prevalent in rose plants. Downy Mildew causes yellow or white patches on the upper surface of the leaves, often accompanied by a downy growth on the underside.

Number of Images:

The dataset comprises a total of 917 photos distributed across the three classes. It is worth noting that the distribution of images among the classes may not be uniform, with some classes having more samples than others. Since the dataset Since the data is not enough that's why wee will create the sample size by doing data augmentation before feeding to the deep learning model.

Data Augmentation:

Data augmentation in plant disease detection involves creating new training samples by applying various transformations to the original images. These transformations can include:

1. **Rotation**: Rotating the image by a certain angle to simulate different orientations of the plant.
2. **Flip**: Flipping the image horizontally or vertically to create mirror images.
3. **Scale**: Resizing the image to a different scale to simulate different distances from the camera.
4. **Translation**: Shifting the image horizontally or vertically to simulate different positions within the frame.
5. **Brightness and Contrast**: Adjusting the brightness and contrast of the image to simulate different lighting conditions.
6. **Noise**: Adding random noise to the image to simulate sensor noise or image imperfections.
7. **Blur**: Applying blur to the image to simulate motion blur or out-of-focus images.

It allows the model to learn from a more diverse set of samples and improve its performance on unseen data.

VGG16:

VGG16 is the name of the convolutional neural network(CNN) architecture that was created at the University of Oxford named after the Visual Geometry Group. It is renowned for being easy to use and efficient at classifying images. VGG16 is distinguished by its deep architecture, which consists of 16 layers, including fully linked, max-pooling, and convolutional layers. The feature of his architecture is given below:

1. **Input layer:** The network receives an input image in RGB channels with a size of 224x224x3.
2. **Convolutional Blocks:** The network is made up of numerous convolutional blocks, each of which has a max-pooling layer after a number of convolutional layers. To maintain the input's spatial dimensions, the convolutional layers employ tiny 3x3 filters with a stride of 1 and the same amount of padding.
3. **Fully Connceted Layers:** Three completely connected layers come after the convolutional layers. There are 4096 neurons in each of the first two fully linked layers, and there are 1000 neurons in the final output layer (equivalent to 1000 ImageNet classes).
4. **Activation Function:** To add non-linearity to the network, the Rectified Linear Unit (ReLU) activation function is employed.
5. **Pooling Layers:** To extract the most notable characteristics and minimize the spatial dimensions of the feature maps, max-pooling layers are employed.
6. **Softmax Activation:** The network's output is transformed into probabilities for each class by the last layer using the softmax activation function.

The ImageNet dataset, a sizable collection of millions of tagged photos from a thousand distinct classes, is used to train VGG16. With its exceptional performance on a range of picture classification tasks, the network has gained popularity as a starting point for transfer learning, a process that uses pre-trained VGG16 models to train on smaller datasets.

ResNet50:

ResNet-50 is a deep convolutional neural network (CNN) architecture that was introduced by Microsoft in 2015. It is renowned for its deep design and capacity to efficiently train very deep neural networks. The following describes this architecture's feature:

1. **Input Layer:** The network receives an input image in RGB channels with a size of 224x224x3.
2. **Convolutional Layers:** To minimize the spatial dimensions of the input, the network begins with a convolutional layer and then moves on to a max-pooling layer.
3. **Residual Blocks:** The numerous convolutional layers that make up each residual block comprise ResNet-50. The use of skip connections, which bypass one or more layers and enable the network to learn residual functions, is the main novelty of ResNet. By addressing the vanishing gradient issue, these skip connections make it possible to train extremely deep networks.
4. **Identity Blocks:** An identity block is a kind of residual block in which the convolutional layers are skipped and the input is added straight to the output. This aids in maintaining the input's original information.
5. **Fully Connected Layers:** A global average pooling layer reduces the spatial dimensions to 1x1, and a fully connected layer with 1000 neurons (equivalent to 1000 ImageNet classes) follows the convolutional layers.
6. **Activation Function:** To add non-linearity to the network, the Rectified Linear Unit (ReLU) activation function is employed.
7. **Softmax Activation:** The network's output is transformed into probabilities for each class by the last layer using the softmax activation function.

Optimizer:
An optimizer is an algorithm that is used during training to change a neural network's weights and biases. By modifying the network parameters in accordance with the gradients of the loss function with respect to the parameters, its main purpose is to minimize the loss function.
Adam: The Adam optimizer, which stands for Adaptive Moment Estimation, is a very popular optimizer. Adam combines the benefits of AdaGrad and RMSProp, two more extensions of stochastic gradient descent. It keeps up a learning rate per parameter adjusted according to the average of the gradients' first and second moments. Adam is a good choice for training deep neural networks because of its variable learning rate and momentum, which allow it to manage noisy data and sparse gradients.
Loss Function:
Loss function is used in machine learning and neural networks to measure how well a model matches the actual labels in the training set with its predictions. For a given input, it measures the difference between the expected and actual output.
Cross-Entropy Loss: In classification problems, cross-entropy loss is frequently used to quantify the difference between expected probability and actual class labels. By modifying the model's parameters (weights and biases) using backpropagation and optimization techniques like Adam or stochastic gradient descent, training aims to reduce this loss.
Early Stopping:
It is a technique used to stop a model from overfitting. It entails keeping an eye on the model's performance on **a** validation set while it is being trained, and halting the process when the performance begins to deteriorate or stops getting better all the while keeping the training loss low. By ending the training process before the model begins to memorize the training data too closely, early stopping aids in the prevention of overfitting. Early stopping can be used to identify whether the model is beginning to overfit by tracking the validation loss during training. This can be done by observing how the model performs on the validation set.

Methodology

To create a useful image dataset, the collected dataset was carefully examined and preprocessed using a variety of techniques. We used Google Colab for implrmrntation purpose where computations executed on the T4 GPU using PyTorch library. Modern CNN models, such as VGG16 and ResNet50, were specifically applied to identify and forecast plant diseases, demonstrating the use of cutting-edge deep learning techniques in agricultural research.

1. **Image Pre-Processing and Data Augmentation:** Recognizing the constraints of our dataset, our primary objective was to identify the class with the most extensive data. Our goal was to enhance the sample size of each class to four times the volume of the class with the maximum number of images.

   Subsequently, we implemented image pre-processing and data augmentation techniques on each image within the class. This approach significantly diversified the dataset, enriching its variety and scope.

Given that both VGG16 and ResNet50 architectures require input images in RGB channels sized 224x224x3, our first step was to resize all images to meet this requirement. Following this, we applied a range of augmentation techniques, including random horizontal flips, random vertical flips, random rotations, and Gaussian blur.

This meticulous approach not only standardized the dataset for compatibility with the models but also introduced a broader range of image variations, enhancing the robustness and effectiveness of our classification models.

2. **Splitting the dataset:** The dataset was meticulously divided into three subsets: training, testing, and validation. The training set comprised 70% of the data, while the testing and validation sets each contained 15% of the data. This partitioning ensured a balanced distribution of data across the subsets, facilitating robust model training, evaluation, and validation.

3. **Model Architecture:**

In both of this two model first we frezze the model parameters then we modified the final layer so that it can fit with our classification task.

Resnet50:

Original final layerLinear(in_features=2048, out_features=1000, bias=True)

Modified final layer Sequential(
 (fc): Linear(in_features=2048, out_features=3, bias=True)
)

Vgg16:

Original final layer Sequential(
 (0): Linear(in_features=25088, out_features=4096, bias=True)
 (1): ReLU(inplace=True)
 (2): Dropout(p=0.5, inplace=False)
 (3): Linear(in_features=4096, out_features=4096, bias=True)
 (4): ReLU(inplace=True)
 (5): Dropout(p=0.5, inplace=False)
 (6): Linear(in_features=4096, out_features=1000, bias=True)
)

Modified final layer Sequential(
 (0): Linear(in_features=25088, out_features=4096, bias=True)
 (1): ReLU(inplace=True)
 (2): Dropout(p=0.5, inplace=False)
 (3): Linear(in_features=4096, out_features=4096, bias=True)
 (4): ReLU(inplace=True)
 (5): Dropout(p=0.5, inplace=False)
 (6): Sequential(
  (0): Linear(in_features=4096, out_features=3, bias=True)
)
)

4. **Other Features:**
    - We use ADAM optimizer and Cross-Entropy loss.
    - Early stopping applied to prevent the model from overfitting.
    - Number of epochs: 16
    - Batch Size: 32

5. **Evaluation Methodology:**
    - Our approach involves evaluating the trained models using the test dataset to calculate key metrics such as accuracy, F1-score, recall, and precision. This will allow us to compare the performance of the VGG16 and ResNet50 architectures effectively.
    - Additionally, we will measure and compare the training times of both the VGG16 and ResNet50 models.
    - Furthermore, we will create visualizations of the training loss and validation loss graphs to facilitate evaluation and comparison between the two models.

Result Analysis

From the table below, we observe that during the training process, ResNet50 and VGG16 took 452.65 seconds and 526.55 seconds, respectively. This indicates that VGG16 required more time compared to ResNet50.

Table 1: Training time of Resnet50 and VGG16 Models

Model Name	Training Time
Resnet50	452.65
VGG16	526.55

Figure 3 and Figure 4 showcase the visualization of the training and validation loss for both models, providing a clear comparison of their performance. We can conclude that overall two of this model is performing good. Upon testing the models with unseen data, we observed that ResNet50 achieved an accuracy of 89.69%, while VGG16 achieved an accuracy of 90.53%. These results indicate that VGG16 outperforms ResNet50 in terms of accuracy. Precison, Recall and F1-Score are common metrics used to evaluate the performance of classification models.

Table 2: Precision, Recall and F1-score for Resnet50

Class Name	Precision	Recall	F1-Score
Black Spot	0.82	0.92	0.87
Downey Mildew	0.99	0.98	0.98
Fresh Leave	0.90	0.79	0.84

Table 3: Precision, Recall and F1-score for VGG16

Class Name	Precision	Recall	F1-Score
Black Spot	0.93	0.83	0.88
Downey Mildew	0.95	0.98	0.96
Fresh leave	0.84	0.91	0.88

Precision is the ratio of correctly predicted positive observations to the total predicted positives, measuring the accuracy of positive predictions. Recall is the ratio of correctly predicted positive observations to all actual positive instances, indicating the model's ability to identify relevant instances. F1 score is the harmonic mean of precision and recall, providing a balance between the two metrics. Regarding precision, ResNet50 achieves 0.82 for black spot detection, while VGG16 achieves a higher precision of 0.93. Conversely, for fresh leaves, VGG16 achieves a precision of 0.84, slightly lower than ResNet50's precision of 0.90. In terms of recall, ResNet50 achieves 0.79, whereas VGG16 achieves a higher recall of 0.91. Despite these differences, both models demonstrate similar F1-Scores. From this observation, we can conclude that VGG16 tends to perform better in terms of precision for detecting both black spot and fresh leaves compared to ResNet50. However, ResNet50 shows a higher recall for black spot detection than VGG16. Figure 5 & 6 is visualizing the results of the predicted output and actual output with images of both of these data.

Discussion

The investigation conducted in this study focused on comparing the performance of two prominent convolutional neural network (CNN) architectures, namely ResNet50 and VGG16, in the detection of rose leaf diseases.

Firstly, the training time analysis revealed that ResNet50 exhibited a shorter training duration of compared to VGG16's training time. This discrepancy suggests that ResNet50 is computationally more efficient, making it a potentially favorable choice in scenarios where training time is a critical factor. Upon testing the models with unseen data, VGG16 demonstrated superior accuracy, achieving an accuracy of 90.53% compared to ResNet50's accuracy of 89.69%. This suggests that VGG16 exhibits better generalization capability and is more adept at capturing the underlying patterns in the data.

Furthermore, the precision-recall-F1 score analysis provided a detailed assessment of the models' performance across different classes of rose leaf diseases. ResNet50 demonstrated higher precison compare to VGG16 for certain classes, indicating its higher precision in positive predictions. However, VGG16 outperformed in terms of recall for certain classes, suggesting its ability to capture a greater proportion of true positive instances.

Overall, the findings suggest that while VGG16 may offer superior accuracy and recall in detecting rose leaf diseases, ResNet50 presents a compelling alternative due to its shorter training time and competitive performance across various evaluation metrics. Researchers and practitioners can leverage these insights to select the most suitable model architecture based on specific requirements and constraints in real-world applications.

Conclusion

Plant disease detection is a critical area of research with profound implications for global agriculture, food security, and environmental sustainability. This study focused on evaluating two different CNN models for detecting plant diseases in rose leaf of Bangladesh, including corn. The results demonstrated that VGG16 performed best compare to ResNet50. The study also highlighted the superior performance of CNN models in plant leaf disease detection.

However, the study is limited to rose leave, and future work will focus on experiments with a combined dataset and a proposed hybrid model. Additionally, efforts will be made to develop a mobile-based application and a web-based interface to facilitate easy and widespread use of the best-performing models for plant disease detection, thereby aiding farmers in timely disease identification and management.

References

[1]. Ref Garima Tripathi and Jagruti Save: “AN IMAGE PROCESSING AND NEURAL NETWORK BASED ROACH FOR DETECTION AND CLASSIFICATION OF PLANT LEAF DISEASES”, Volume 6, Issue 4, April (2015), pp. 14-20.

[2]. Ahmed, K., Shahidi, T. R., Alam, S. M. I., & Momen, S. (2019, December). Rice leaf disease detection using machine learning techniques. In *2019 International*

Conference on Sustainable Technologies for Industry 4.0 (STI) (pp. 1-5). IEEE.

[3]. Nuanmeesri, S. (2021). A hybrid deep learning and optimized machine learning approach for rose leaf disease classification. *Engineering, Technology & Applied Science Research*, *11*(5), 7678-7683.

[4]. Rajbongshi, A., Sarker, T., Ahamad, M. M., & Rahman, M. M. (2020, October). Rose diseases recognition using MobileNet. In *2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)* (pp. 1-7). IEEE.

[5]. Zimao, L., Liandong, L., Meng, X., Jun, T., & Yue, Z. (2021). Detection of rose diseases and insect pests based on deep learning. *Journal of Chinese Agricultural Mechanization*, *42*(8), 169.