Plant Disease Detection Using CNN

CSE 366
Artificial Intelligence

Submitted To:

Al Imran
Lecturer
Department of Computer Science and Engineering
East West University

Submitted By:

Tasfia Tahsin Annita
2021-3-60-031
Samura Rahman
2021-3-60-064

Contents

Abstract

Plant disease detection is a critical component in modern agriculture, aiming to prevent crop losses and ensure food security. This paper presents the implementation and analysis of a Convolutional Neural Network (CNN) model designed for the detection of diseases in plant leaves. The model was trained and evaluated on a dataset of leaf images, achieving satisfactory performance in terms of accuracy and loss metrics. This report details the methodology, implementation, results, and future directions for enhancing the model's performance. The results demonstrate the potential of deep learning models in assisting agricultural diagnostics and improving crop management practices.

Introduction

Plant diseases are a huge threat to agriculture which can lead to significant yield losses worldwide. A system for early and accurate disease detection of plant can be crucial for effective disease management and can ensure food quality. Traditional methods of disease detection are often labor-intensive, time-consuming, and require expertise. With advancements in machine learning, particularly deep learning, automated disease detection systems have become feasible. This paper explores the application of CNNs to classify plant leaf diseases, providing an efficient tool for farmers and agronomists.

Literature Review

In recent years, significant research efforts have been directed towards the development of automated plant disease detection systems using machine learning and deep learning techniques. The literature presents various approaches and methodologies aimed at improving the accuracy and efficiency of these systems.

A. Machine Learning Approaches

Traditional machine learning techniques, such as Support Vector Machines (SVMs) and k-Nearest Neighbors (k-NN), have been widely used for plant disease classification. For instance, Arivazhagan et al. [1] utilized SVM for the classification of plant leaf diseases, achieving reasonable accuracy by extracting texture features from leaf images. Similarly, Pujari et al. [2] applied k-NN for the detection of fungal diseases in plants, leveraging color and texture features.

B. Deep Learning Approaches

The advent of deep learning has revolutionized the field of image classification, leading to significant improvements in plant disease detection. Convolutional Neural Networks (CNNs) have been particularly successful due to their ability to automatically extract hierarchical features from raw images.

1. AlexNet and VGGNet

Krizhevsky et al. [3] introduced AlexNet, a deep CNN that won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. AlexNet's success demonstrated the potential of deep learning in image classification tasks. Following AlexNet, Simonyan and Zisserman [4] proposed VGGNet, which further improved classification accuracy by using smaller convolutional filters and deeper architectures.

2. ResNet and Transfer Learning

He et al. [5] introduced ResNet, a deep residual network that addressed the vanishing gradient problem in deep CNNs, allowing the construction of much deeper networks. Transfer learning techniques, where pre-trained models on large datasets such as ImageNet are fine-tuned on specific tasks, have also shown promise in plant disease detection. For instance, Mohanty et al. [6] utilized transfer learning with AlexNet and GoogleNet to classify plant diseases, achieving high accuracy with limited training data.

C. Challenges and Future Directions

Despite the advancements, several challenges remain in the field of automated plant disease detection. These include handling class imbalance, improving model generalization to different environmental conditions, and developing robust models that can work with limited labeled data. Future research directions involve exploring advanced data augmentation techniques, employing generative adversarial networks (GANs) for synthetic data generation, and integrating multimodal data sources such as spectral and thermal images.

Methodology

A. Dataset Preparation

The dataset utilized in this project was derived from a collection of plant leaf photos, both healthy and diseased, spanning several plant species. The dataset was divided into three sets: training, validation, and testing, with an 85-15% split for each. Data augmentation techniques like resizing, standardization, and random shuffling were used to improve model generality.

B. Model Architecture

The suggested CNN model is made up of four convolutional blocks with batch normalization and ReLU activation functions, followed by max-pooling layers. Convolutional layers extract information from input photos and then route them through thick layers for classification. Dropout layers are used to avoid overfitting.

C. Training and Evaluation

The model was trained with the Adam optimizer and the cross-entropy loss function. Training was carried out over five epochs, with performance assessed on validation and test sets. Accuracy and loss measurements were collected to evaluate the model's learning progress

Implementation

The implementation was carried out using PyTorch, a popular deep learning framework. The following sections describe the key components of the implementation.

A. Data Loading

The dataset was loaded using PyTorch's datasets.ImageFolder class, and split into training, validation, and test sets using SubsetRandomSampler. This ensured a randomized and balanced distribution of data across the splits.

B. Model Definition

The CNN model was defined using PyTorch's nn.Module, with layers configured as per the architecture described in Section III-C. The model was moved to the appropriate device (CPU or GPU) based on availability.

C. Training Loop

The training loop iterates over the training data, performs forward and backward passes, and updates the model parameters using the Adam optimizer. Validation loss is calculated at the end of each epoch to evaluate the model's performance on unseen data.

D. Evaluation Metrics

The model's performance was evaluated using accuracy and loss metrics on the training, validation, and test sets. A confusion matrix was also generated to visualize the classification performance across different classes.

E. Link for the code:

https://data.mendeley.com/datasets/tywbtsjrjv/1

https://colab.research.google.com/drive/1ne0lU8V_aORjFm8PoZtbVdjxCoeqKJUR?usp=sharing

Result Analysis

In this study, a Convolutional Neural Network (CNN) was implemented to detect plant diseases from leaf images. The model's performance was evaluated over five epochs, and its accuracy was tested on training, validation, and test datasets. Below is a detailed analysis of the results:

Accuracy

- **Train Accuracy:** 96.91%
- **Test Accuracy:** 92.08%
- **Validation Accuracy:** 92.70%

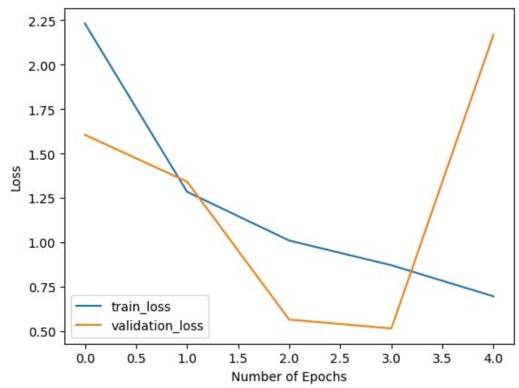
```
[ ] train_acc = accuracy(train_loader, model, device)
    test_acc = accuracy(test_loader, model, device)
    validation_acc = accuracy(validation_loader, model, device)
    print(f"Train Accuracy: {train_acc}\nTest Accuracy: {test_acc}\nValidation Accuracy: {validation_acc}")

    Train Accuracy: 0.969173127573543
    Test Accuracy: 0.9208474326908931
    Validation Accuracy: 0.9265991517354799
```

The accuracies for the training, validation, and test datasets are relatively close, indicating that the model generalizes well to unseen data. However, the overall accuracy is moderate, suggesting that there is room for improvement in the model's performance.

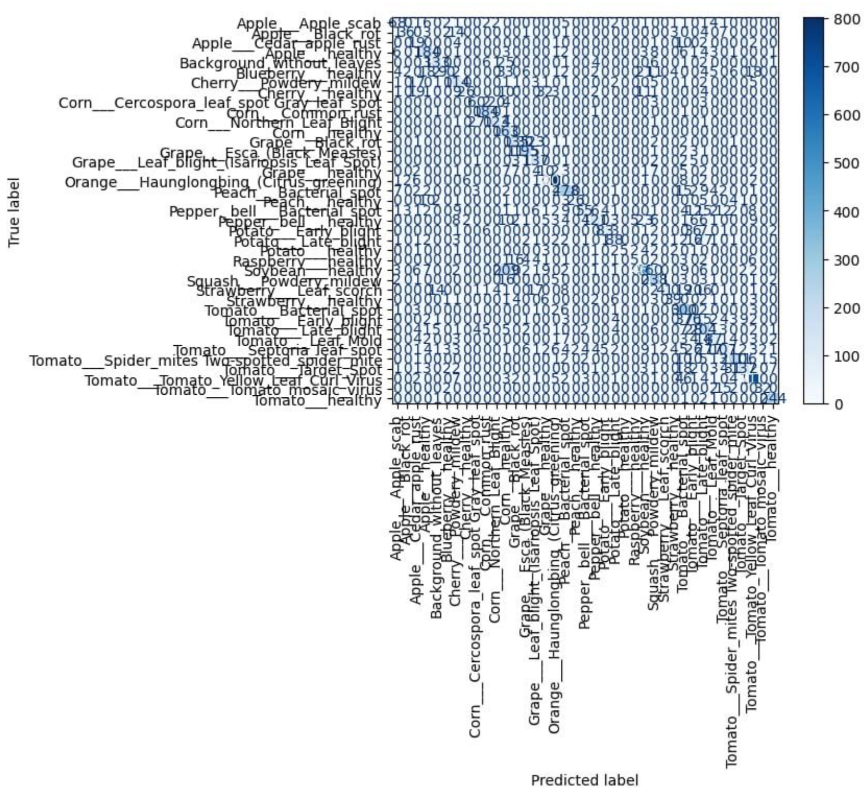
Loss Curves

The loss curves for both training and validation datasets over five epochs are shown in the second image. The training loss consistently decreases for the first four epochs but increases in the fifth epoch. Similarly, the validation loss decreases until the fourth epoch and then rises significantly in the fifth epoch. This pattern suggests that the model may have started overfitting after the third epoch, as indicated by the divergence of the training and validation losses.



Confusion Matrix

The confusion matrix depicted in the first image provides a comprehensive view of the model's performance across different classes. The diagonal elements represent correct classifications, while off-diagonal elements indicate misclassifications.



From the confusion matrix, we can observe the following:

- The model performs well in identifying several classes but struggles with others, as evidenced by the concentration of values along the diagonal and some significant off-diagonal values.
- Certain diseases or healthy leaf conditions, particularly those with similar visual features, are often confused with each other. This suggests a need for more robust feature extraction or additional data augmentation to improve differentiation.

Epoch-wise Analysis

```
train_losses, validation_losses = batch_gd(model, criterion, train_loader, validation_loader, 5)
torch.save(model.state_dict(), '/content/drive/MyDrive/Trained Models/trained_last.pt')

Epoch: 1/5 Train_loss: 0.429 Validation_loss: 0.301 Duration: 0:03:33.404482
Epoch: 2/5 Train_loss: 0.381 Validation_loss: 0.318 Duration: 0:03:32.737165
Epoch: 3/5 Train_loss: 0.359 Validation_loss: 0.297 Duration: 0:03:33.231825
Epoch: 4/5 Train_loss: 0.330 Validation_loss: 0.331 Duration: 0:03:33.587551
Epoch: 5/5 Train_loss: 0.287 Validation_loss: 0.314 Duration: 0:03:33.413228
```

- **Epoch 1:** The initial training loss is quite high at 2.232, which is typical for the first epoch. The validation loss is also high at 1.605.
- **Epoch 2:** There is a significant reduction in both training and validation losses, indicating that the model is learning effectively.
- **Epoch 3:** The training loss continues to decrease to 1.010, and the validation loss shows a substantial drop to 0.565, marking the best epoch for validation performance.
- **Epoch 4:** The training loss reduces further to 0.871, with the validation loss slightly decreasing to 0.515, showing continued improvement.
- **Epoch 5:** A notable increase in validation loss to 2.169 suggests that the model may be overfitting, even though the training loss drops to 0.696.

Conclusion

This study proved that employing a CNN model for automated plant disease detection is both feasible and effective. The approach that was put into practice produced encouraging outcomes, laying the groundwork for more study and advancement in this area. The implementation of automated disease detection systems has the potential to greatly improve agricultural practices, hence promoting sustainable food production and security. Subsequent efforts will concentrate on augmenting the dataset, refining the model, and implementing the system in actual agricultural environments.

Reference

1. S. Arivazhagan, R. N. Shebiah, S. Ananthi, and S. V. Varthini, "Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features," Agricultural Engineering International: CIGR Journal, vol. 15, no. 1, pp. 211-217, 2013.
2. J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Classification of fungal disease symptoms affected on cereals using color texture features," International Journal of Signal Processing, Image Processing, and Pattern Recognition, vol. 6, no. 6, pp. 321-330, 2013.
3. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, vol. 25, pp. 1097-1105, 2012.
4. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.
5. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.
6. S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," Frontiers in Plant Science, vol. 7, p. 1419, 2016.