Research on Java Apple Disease Identification Using Convolutional Neural Networks and Transfer Learning

Cha Xinrong, Jiang Ying, Cheng Yong, Wang Ziying

Yunnan Normal University, Kunming, 650500, Yunnan, China

Abstract: *Java apple (Syzygium samarangense), also known as wax apple, is a tropical fruit prized for its unique taste and nutritional value. However, its cultivation faces serious threats from pests and diseases, leading to reduced yields and economic losses. Traditional disease detection methods rely on manual inspection, which is time-consuming, labor-intensive, and prone to errors. This study proposes a classification method using convolutional neural networks (CNN) and transfer learning for efficient and accurate disease identification in Java apples. We developed a CNN model with three convolutional layers and two fully connected layers, trained on our dataset.Additionally, we fine-tuned pre-trained models—VGG16, InceptionV3, and ResNet50—and optimized their training cycles and optimizer configurations. The experimental results showed that the InceptionV3 model achieved the highest classification accuracy at 99.37%, while ResNet50 and VGG16 had test accuracies of 98.89% and 97.23%, respectively. These findings indicate that combining transfer learning with CNNs significantly enhances the accuracy of Java apple disease detection, providing an effective and scalable solution that supports crop health monitoring.*

Keywords: *Convolutional Neural Networks (CNN); disease detection; transfer learning; VGG16; InceptionV3; ResNet50*

1. Introduction

Java apple (Syzygium samarangense) cultivation currently faces significant threats from various pests and diseases, particularly Anthracnose disease and Fruit Rot, which lead to decreased production and substantial economic losses. Traditional detection methods predominantly rely on manual inspection, a process that is not only time-consuming but also prone to human error. Given the rapid spread of these diseases and the need for early intervention, there is a critical demand for more efficient and accurate disease identification methods.

Recent advancements in agricultural technology and high-resolution imaging have positioned computer vision as a promising alternative for pest and disease detection in crops[1]. By leveraging image processing and machine learning, computer vision systems can automatically identify and classify visual patterns associated with different diseases. Among these, deep learning techniques, especially convolutional neural networks (CNNs), have demonstrated remarkable accuracy in image recognition tasks across various domains[1]. Despite their success in other crops like rice and wheat, deep learning models remain underutilized in the context of Java apple disease detection[2]. The scarcity of large, labeled datasets and the focus of research on more common crops contribute to this gap.

Developing an effective detection system for Java apple diseases could significantly enhance crop yield, reduce economic losses, and promote sustainable farming practices. This study aims to bridge this gap by developing a CNN architecture with three convolutional layers and two fully connected layers, and by applying transfer learning models, including ResNet50, VGG16, and InceptionV3, to identify and classify diseases in Java apples.Through extensive experiments and evaluations, we seek to determine the most effective model configurations, thereby providing scalable and reliable solutions for real-time monitoring and management of crop health in Java apple orchards.

2. Literature Review

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have significantly improved image-based disease detection in agriculture[1]. Studies have shown the effectiveness of architectures like VGG16 and ResNet50 for plant disease classification, particularly in crops such as rice, wheat, and tomato[4][6]. These models have demonstrated high accuracy in identifying various plant diseases by learning complex visual patterns directly from images. For instance, VGG16 has been successfully used in classifying leaf diseases in tomatoes, while ResNet50 has shown robustness in detecting wheat diseases, as reported in several studies.

Transfer learning, which leverages pre-trained models such as ResNet50, VGG16, and InceptionV3, has proven particularly valuable for agricultural datasets, which are often limited in size[3]. By using models pre-trained on large datasets like ImageNet, transfer learning enables enhanced model performance without the need for large, labeled training datasets[3]. This approach has been shown to significantly reduce the training time and improve the accuracy of disease detection models, even with limited agricultural data.

Additionally, data augmentation techniques, including rotation, flipping, and contrast adjustments, have been widely adopted to improve model generalization in the context of small datasets. These techniques create variations of existing images, effectively increasing the size of the training set and helping models learn more robust features[5]. Studies have consistently shown that applying such augmentations leads to better performance in plant disease detection tasks[7].

Building on these advancements, this study applies CNNs and transfer learning models to the specific context of Java apple disease detection. By utilizing a balanced dataset and strategic data augmentation, this research aims to ensure robust and accurate disease classification. Unlike previous studies focused on more common crops, this work addresses the gap in the literature regarding the application of these advanced techniques to Java apple disease detection, promising a scalable solution for effective crop monitoring.

3. Methodology

3.1 Dataset Description

The dataset used in this study comprises 1,500 images of Java apples (Syzygium samarangense), evenly distributed across three categories: Fresh (500 images), Rotting (500 images), and Anthracnose (500 images). These images were sourced from various origins to ensure diversity and representativeness. Fresh Java apple images were collected from supermarkets, market stalls, and online stores, reflecting the appearance of the fruit in different environments. Rotting Java apple images were captured at various stages of decay, sourced either from field experiments or collected online, and were carefully selected to ensure representativeness. The Anthracnose images were sourced from Google and relevant academic literature, initially comprising 51 images, which were then expanded to 500 images using data augmentation techniques such as mirroring, rotation, and contrast enhancement to balance the dataset categories.

All images were resized to 224x224 pixels, maintaining the original aspect ratio to ensure consistency. Additionally, to improve the model's generalization ability and prevent overfitting, we applied further data augmentation, including image flipping, scaling, and noise addition. These data augmentation techniques increase the diversity of the images, enabling the model to perform better when encountering unseen data and effectively reducing the risk of overfitting during training, thereby enhancing the model's stability and predictive accuracy. As shown in Figure 1, an example data enhancement diagram demonstrates examples of the image enhancement techniques used in this study.

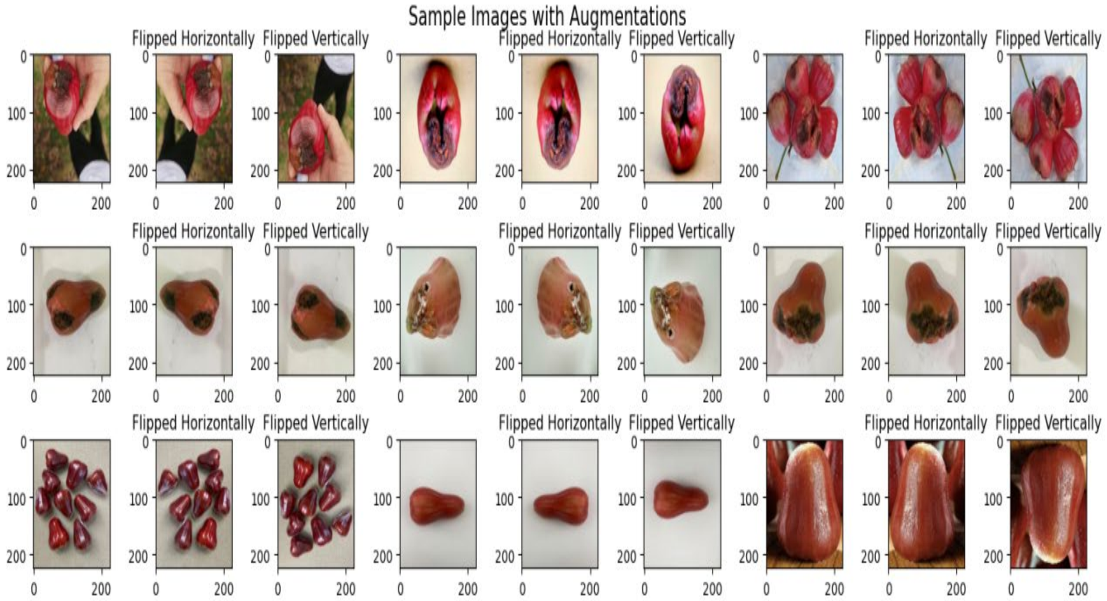


Figure 1: Example Data Enhancement Diagram

3.2 Proposed CNN Model

The proposed CNN model consists of three convolutional layers. The first layer has 128 filters of size 5x5, followed by a second layer with 64 filters of size 3x3 and L2 regularization (0.00005), and a third layer with 32 filters of size 3x3, also using L2 regularization. L2 regularization is applied to the convolutional layers to prevent overfitting by penalizing large weights. ReLU activation is applied after each convolution to enhance nonlinear processing. Max-pooling layers follow each convolutional layer to reduce the spatial dimensions while preserving essential features. As shown in Figure 2, the architecture of the proposed CNN model includes these convolutional and pooling layers, followed by fully connected layers.

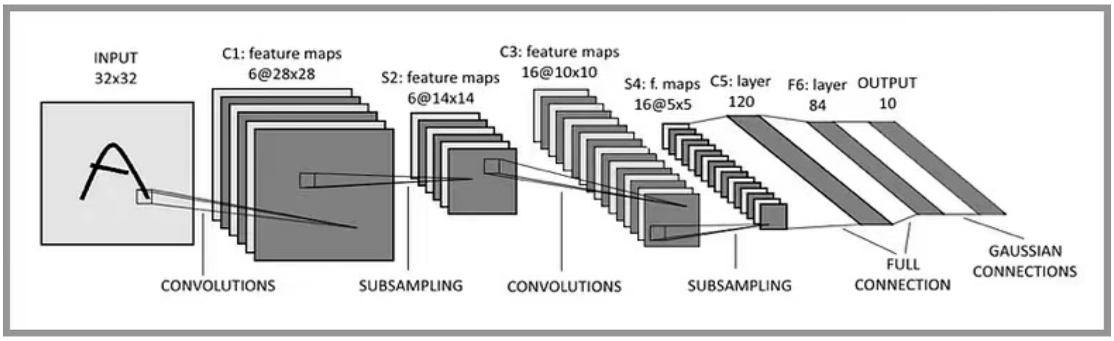


Figure 2:cnn-based construction modeling.

The network incorporates batch normalization to improve training stability and speed up convergence. The feature maps are then flattened and passed to a fully connected layer with 224 neurons and ReLU activation. A Dropout layer with a 50% rate is included to prevent overfitting. The final output layer has 3 neurons, corresponding to the three classes (Fresh, Rotting, and Anthracnose), and uses a softmax function to convert features into probability distributions for classification.

The choice of optimizer is critical in CNN models as it influences both training efficiency and final performance. We employed grid search to select optimal parameters, testing combinations of SGD, Adam, and RMSprop optimizers across 20-50 training epochs. Cross-validation was used to ensure robust results by evaluating performance on different data subsets.

3.3 Proposed Transfer Learning Models

In our study, alongside the proposed convolutional neural network (CNN), we utilized transfer learning models including ResNet50, VGG16, and InceptionV3 to compare their accuracy with our CNN model. Transfer learning is highly effective in computer vision tasks, allowing models to leverage knowledge gained from pre-training on large-scale datasets to improve performance on new tasks. We utilized pre-trained models (ResNet50, VGG16, InceptionV3) by replacing their original classification layers with new ones tailored to our specific task of Java apple disease detection. The weights for these models were initialized using ImageNet training results, and all layers were included in the study to ensure comprehensive learning. Table 1 lists the common hyperparameters used for these models.

Table 1: Hyperparameters used in the base models (ResNet50, VGG16, InceptionV3).

Parameter	ResNet50	VGG16	InceptionV3
Input image	(224,224,3)	(224,224,3)	(224,224,3)
Weight	Initialized to ImageNet	Initialized to ImageNet	Initialized to ImageNet
Optimizer	Adam	Adam	Adam
Loss Function	Sparse categorical cross entropy	Sparse categorical cross entropy	Sparse categorical cross entropy
Classifier	Softmax	Softmax	Softmax
Epochs	30	30	30
Dropout rate	0.2	0.2	0.2

3.3.1 VGG16 model

In this study, the pre-trained VGG16 'ImageNet' model was customized for the classification of 'Anthracnose Disease', 'Fruit Rot Disease', and 'Java Apple Healthy'. The original three fully-connected layers were replaced with a global average pooling layer, a 1024-dimensional dense layer with ReLU activation, and a final dense layer with three output units. This customized model retains 15 trainable layers and approximately 30.45 million parameters. To optimize performance, grid search was used to fine-tune the model, involving the unfreezing of the top four layers for weight updates. Additional layers were added to enhance overfitting resistance and improve generalization.

3.3.2 InceptionV3

The InceptionV3 architecture leverages inception modules, which use parallel convolutional layers with various kernel sizes to capture features at different scales[5]. This model reduces the number of parameters using dimensionality reduction techniques, thus achieving high accuracy in image recognition tasks while maintaining computational efficiency.

3.3.3 Resnet50

ResNet50 is a 50-layer deep convolutional neural network that uses residual connections to facilitate the training of very deep networks[6]. This architecture is particularly effective at overcoming the vanishing gradient problem, allowing for more straightforward optimization of deep models and achieving impressive performance in image classification tasks.

3.4 Performance Metric

In this project, we measure the model's performance using accuracy, a suitable metric given that the dataset is evenly distributed across classes. Accuracy is calculated using the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative.

3.5 Hyperparameter Tuning

To optimize model performance, we tuned two key hyperparameters: learning rate and batch size. The learning rate controls the magnitude of updates to the model weights with each step, while the batch size determines the number of samples processed before updating the model. We tested learning rates of 0.01, 0.001, and 0.0001, along with batch sizes of 32 and 64, across six experiments using Keras Tuner. After training, the model with the highest validation accuracy was further evaluated on a test set, focusing on both the custom CNN and the transfer learning models. The impact of these hyperparameters on model accuracy is thoroughly analyzed to determine the optimal configuration.

4. Experiment Results

In the following four images, the training and validation losses during hyperparameter tuning are shown. Generally, both losses decrease over time. However, some trials do not converge and remain flat throughout the training period. This issue is particularly evident in the transfer learning models at a learning rate of 0.01. On the other hand, the training and validation losses were minimized with a learning rate of 0.001.

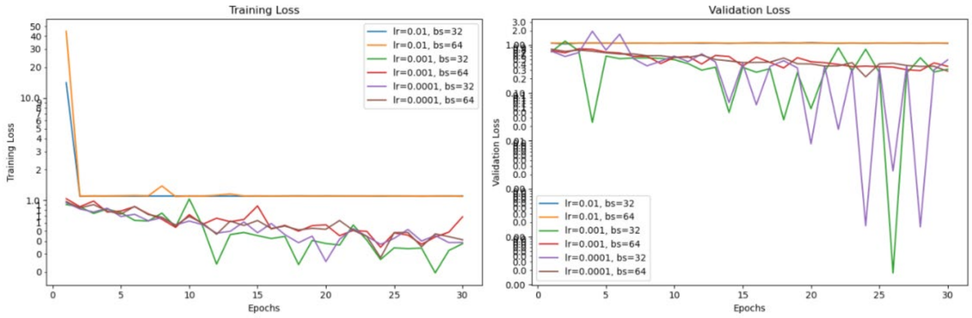


Figure 3: CNN loss.

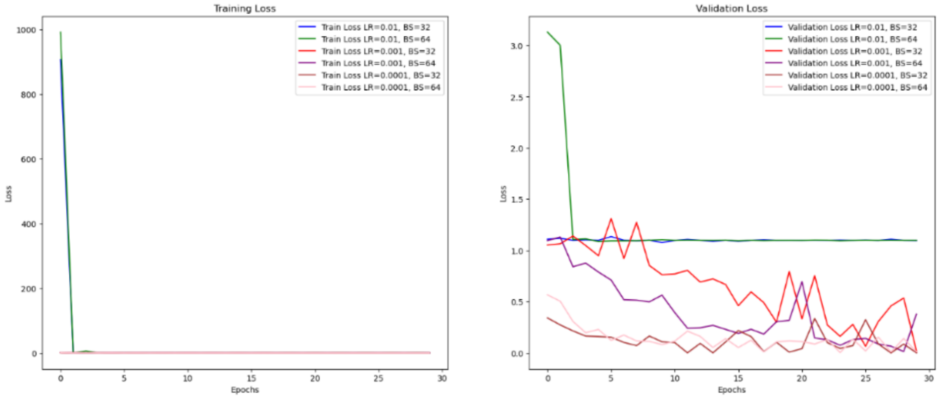
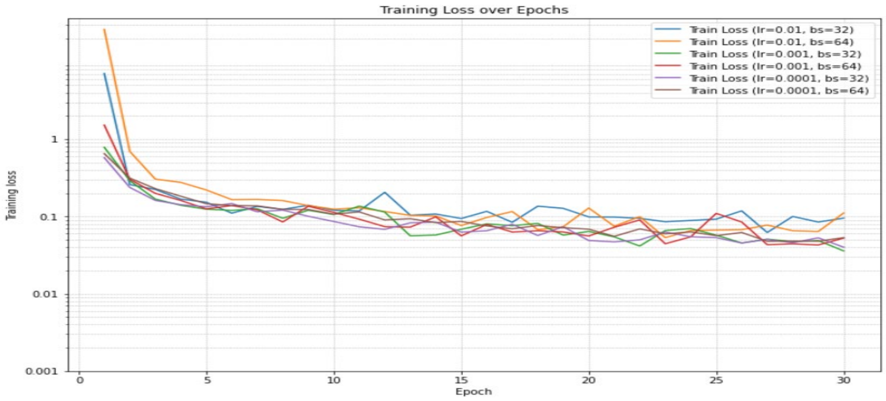


Figure 4: VGG16 loss.



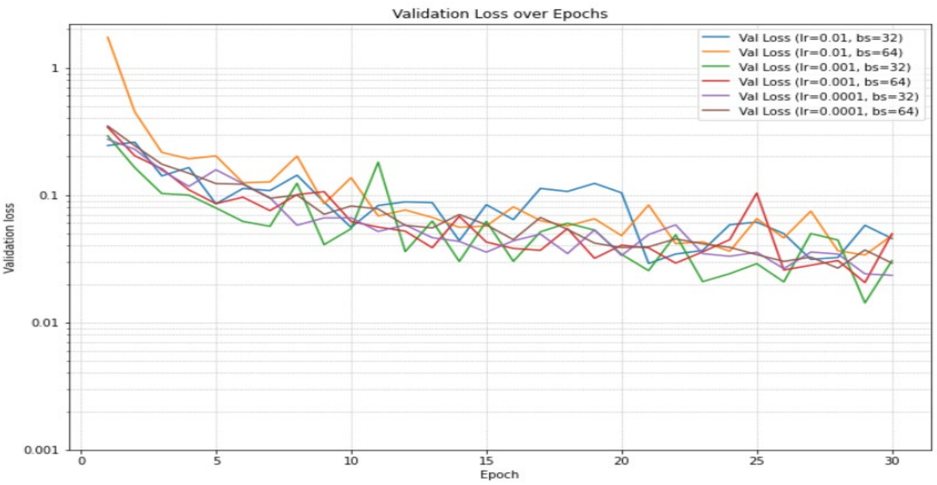


Figure 5: InceptionV3 loss.

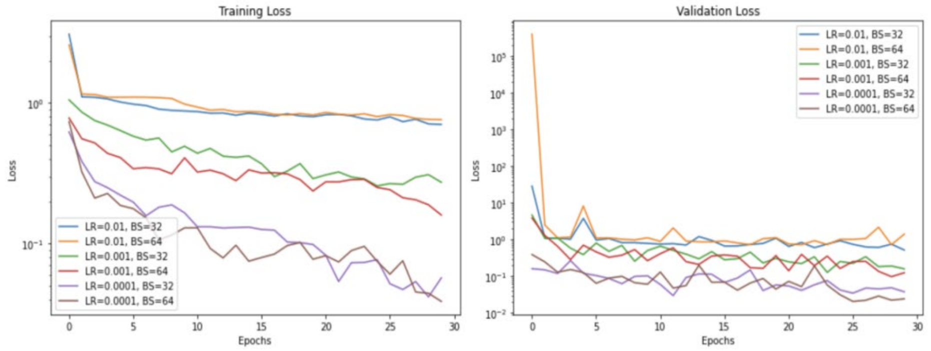


Figure 6: ResNet50 loss.

Table 2: Best validation accuracy of image classification models for different learning rates and batch sizes used during the hyperparameter tuning process.

Trial	Learning Rate	Batch Size	Validation Accuracy (%)			
			CNN	VGG16	InceptionV3	Resnet50
1	0.01	32	28.12	34.22	98.91	78.33
2	0.01	64	34.37	33.45	98.32	69.00
3	0.001	32	**91.47**	82.53	98.87	96.66
4	0.001	64	68.75	77.84	98.82	95.33
5	0.0001	32	81.25	**96.53**	**99.37**	**98.66**
6	0.0001	64	75.30	92.94	99.12	97.57

Table 2 summarizes the optimal validation accuracies of the proposed CNN and transfer learning models using different learning rates and batch sizes during hyperparameter tuning. In general, a smaller batch size (32) leads to better validation accuracy compared to a larger batch size (64). For the transfer learning models, a smaller learning rate generally results in better validation accuracy. This is also true for the CNN model, where a learning rate of 0.001 produced the highest validation accuracy. It is worth noting that for all transfer learning models with a learning rate of 0.01, except for the InceptionV3 model, the accuracies significantly dropped compared to other learning rates. This is likely due to the failure of these models to converge during training, as shown in the loss curves where the loss remains flat instead of decreasing over time.

Table 3 shows the highest validation accuracy for each model from the hyperparameter tuning process, while the accuracy values and their corresponding hyperparameters are summarized in Table 2. From the hyperparameter tuning process, the best CNN model achieved a learning rate of 0.001, a batch size of 32, and a validation accuracy of 91.47%. For the other transfer learning models, the highest validation accuracy was achieved with a learning rate of 0.0001 and a batch size of 32. The validation accuracies for InceptionV3, ResNet50, and VGG16 were 99.37%, 98.66%, and 96.53%, respectively. Based on these validation accuracy scores, we find that InceptionV3 provided the best validation accuracy, followed by ResNet50, VGG16, and the proposed CNN network. Additionally, the best-performing models were tested on the test dataset to obtain the test accuracy. Overall, the test accuracy is comparable to the validation accuracy, although the ranking differs slightly. Both ResNet50 and InceptionV3 achieved a detection accuracy of around 98.7%, followed by VGG16 with a detection accuracy of 97.23%.

Table 3 :Validation accuracy and test accuracy of image classification models with their most optimal hyperparameters.

Model	Hyperparameters	Validation Accuracy (%)	Test Accuracy (%)
CNN	Learning rate: 0.001 Batch size: 32	91.47	94.53
VGG16	Learning rate: 0.0001 Batch size: 32	96.53	**97.23**
Inceptionv3	Learning rate: 0.0001 Batch size: 32	**99.37**	98.59
Resnet**50**	Learning rate: 0.0001 Batch size: 32	98.66	**98.89**

5. Conclusion

This study proposed a Java apple disease identification method based on Convolutional Neural Networks (CNN) and transfer learning and validated its effectiveness by comparing it against various deep learning models. We conducted detailed experimental analyses using a custom CNN model as well as three pre-trained transfer learning models (VGG16, InceptionV3, ResNet50). The results showed that InceptionV3 performed the best on the validation set, achieving a validation accuracy of 99.37%, while ResNet50 and VGG16 achieved test accuracies of 98.89% and 97.23%, respectively. The custom CNN model also achieved a validation accuracy of 91.47% after optimization.

These findings indicate that transfer learning models, particularly InceptionV3 and ResNet50, have significant advantages in handling the Java apple disease classification task, effectively addressing the common issue of data scarcity in agricultural image classification. Our research not only provides reliable technical support for the early detection of Java apple diseases but also lays a foundation for the development of smart agriculture[7].

Future research could focus on expanding the range of detectable diseases, refining model architectures to further improve accuracy and efficiency, and integrating these models into Internet of Things (IoT) systems for real-time monitoring and management of Java apple orchards. Specifically, exploring the use of edge computing devices for on-site disease detection and creating a comprehensive database of Java apple diseases could significantly enhance the practical application of these models in precision agriculture.

References

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
[2] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems (NIPS), 25, 1097-1105.
[3] Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), 1345-1359.
[4] Simnyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. International Conference on Learning Representations (ICLR).
[5] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826.
[6] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778.
[7] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in Plant Science, 7, 1419.