Interpretable Deep Learning Framework for Early Classification of Tomato and Grapevine Leaf Diseases

Geethika Ramaiah Edara[●], Aluru Ranganadha Reddy*[●]

Department of Bioinformatics, Vignan's Foundation for Science, Technology and Research, 522213 Guntur, India

* Correspondence: Aluru Ranganadha Reddy (hodbioinformatics@vignan.ac.in)

Abstract: The integration of artificial intelligence (AI) in precision agriculture has facilitated significant advancements in crop health monitoring, particularly in the early identification and classification of foliar diseases. Accurate and timely diagnosis of plant diseases is critical for minimizing crop loss and enhancing agricultural sustainability. In this study, an interpretable deep learning model—referred to as the Multi-Crop Leaf Disease (MCLD) framework—was developed based on a Convolutional Neural Network (CNN) architecture, tailored for the classification of tomato and grapevine leaf diseases. The model architecture was derived from the Visual Geometry Group Network (VGGNet), optimized to improve computational efficiency while maintaining classification accuracy. Leaf image datasets comprising healthy and diseased samples were employed to train and evaluate the model. Performance was assessed using multiple statistical metrics, including classification accuracy, sensitivity, specificity, precision, recall, and F1-score. The proposed MCLD framework achieved a detection accuracy of 98.40% for grapevine leaf diseases and a classification accuracy of 95.71% for tomato leaf conditions. Despite these promising results, further research is required to address limitations such as generalizability across variable environmental conditions and the integration of field-acquired images. The implementation of such interpretable AI-based systems is expected to substantially enhance precision agriculture by supporting rapid and accurate disease management strategies.

Keywords: Plant disease detection; Deep learning; Convolutional neural network; Artificial intelligence; Leaf disease

1 Introduction

Plant diseases have been known to threaten agricultural productivity and subsequently lead to the unavailability of foodstuffs in the absence of early detection. Rice and maize, however, are cores of agricultural productivity and food accessibility. Management and mitigation of plant diseases in agriculture depend on early detection as well as forecasting. In rural areas in developing countries, agricultural specialists still rely on visual diagnosis for the identification of plant diseases. The traditional method calls for expert monitoring quite often; thus, it becomes expensive for large-scale farms [1]. On the other hand, farmers in remote locations have to travel long distances to get expert consultation from an expert which increases costs and delays timely disease management. Hence, the traditional method has limitations in scope and scalability.

Therefore, research on automated plant disease detection is significant. Automation can efficiently survey vast agricultural landscapes and promptly detect symptoms of diseases in plant foliage [2]. Systems for identifying plant diseases that are rapid, automatic, cost-effective, and reliable are needed. Many studies have used classifiers, such as K-nearest neighbor (KNN), support vector machine (SVM), Fisher linear discriminant (FLD), artificial neural network (ANN) and random forest (RF), to classify plant images into healthy and infected. The leaves show the earliest indications of plant diseases [3]. Feature extraction methods such as the seven Hu invariant moments, scale-invariant feature transform (SIFT), Gabor transform, global-local singular value decomposition, and sparse representation have been employed in classical approaches for segmenting infected areas. Such a design in hand-crafted features demands an expert intervention that can be subjective and create problems in feature selection. Moreover, under tricky field conditions, existing segmentation methods rarely identify diseased leaf regions with sufficient accuracy, making the automatic detection of plant diseases challenging. Deep learning, especially CNNs,

has recently shown great success in tackling these problems. CNNs are particularly effective in image categorization for both small- and large-scale applications [4]. For example, Mohanty et al. [5] achieved a stunning accuracy of 99.35% using a trained CNN model to distinguish between 14 crop species and 26 diseases. Using deep CNNs, Ma et al. [6] accurately identified downy mildew, anthracnose, powdery mildew, and target leaf spots on cucumbers with an accuracy of 93.4%. Although these studies show excellent results, the image datasets used in them are hardly diverse-spotting under controlled laboratory conditions instead of agricultural settings. CNNs have not exhibited much progress in identifying disease symptoms on leaves in field images with complex backgrounds. Because of the large number of trainable parameters, CNNs require large datasets with labeled examples, which are difficult to collect. Despite these limitations, deep learning research has shown promise. To address the limitations of classical CNNs, transfer learning techniques are useful. Fine-tuning only the final classification layers of pretrained networks is one such approach [7].

This work employs deep transfer learning within CNN architectures toward enhanced detection of slight symptom expressions of diseases with a lesser computation burden. In the proposed approach, the pretrained CNN module plays the role of basic feature extraction, while an auxiliary module is tasked with multi-scale feature representations for improved detection. Particularly, the addition of the Inception module enhances the VGGNet architecture. The VGGNet architectural modification includes changing the last convolutional layer to a $3 \times 3 \times 512$ layer and incorporating batch normalization and Swish activation functions in place of the Rectified Linear Unit (ReLU). Two Inception modules follow that extract multi-scale features from convolutional outputs. Global average pooling replaces fully connected layers for feature map dimensionality reduction. A fully connected Softmax layer is added at the end and customized to the number of disease classes, thus making INC-VGGN, a variant CNN for plant disease classification.

Identification of plant diseases is very important in maintaining agricultural productivity and economic stability. Early diagnosis of diseases in plants reduces crop loss due to diseases, assures food supply for the growing population, and helps preserve farmers' revenues. The following part describes some recent applications of machine learning for identifying plant diseases. Ferentinos et al. [8] used the CNN architectures AlexNet, AlexNetOWTBn, GoogleNet, Overfeat, and Visual Geometry Group (VGG) to classify plant diseases. The models were trained on 87,848 images of 25 plant species and 58 plant-disease pairs. The best model classified damaged and healthy plants with an accuracy of 99.53%. Mohanty et al. [5] used 54,306 images of diseased and healthy leaves in an open-source dataset. Their model used deep CNN (AlexNet and GoogleNet) for multicategory classification of 14 crops and 26 diseases with almost 99.35% accuracy on a different test set. Mehedi et al. [9] pre-trained EfficientNetV2L, MobileNetV2, and ResNet152V2 for transfer learning. The model diagnosed 38 leaf diseases across 14 plant species using data from Kaggle. The best-performing model achieved an accuracy of 99.63%, which was the case for EfficientNetV2L. Local Interpretable Model-Agnostic Explanations (LIME), a tool for explainable artificial intelligence (XAI), helped explain the predictions made by the models. Ramesh et al. [10] classified healthy and diseased leaves using the RF classifier. After creating a dataset and extracting features using Histogram of Oriented Gradients (HOG), the classifier was trained and classification was conducted. The model predicted approximately 70% accuracy on 160 images of papaya leaves.

Jasim and Al-Tuwaijari [11] showed that deep learning models are superior to the typical machine learning algorithms in the detection and categorization of early plant diseases. For this, 20,636 Plant Village images were used, with tomato, pepper, and potato crops selected for their economic significance. A model with an accuracy of 98.03% was developed based on CNN, which can be further enhanced by adding more training data. Harakannanavar et al. [12] applied machine learning with image processing to identify diseases in tomatoes. A commendable strategy was proposed for the preliminary detection of diseases by using SVM, KNN, and CNN as classifiers. The model achieved accuracies of 88%, 97%, and 99.6% with each approach, respectively. The CNN-based model proposed by Benito Fernández et al. [13] was tuned to XAI methods such as LIME (2016), SHapley Additive exPlanations (SHAP) (2017), and Gradient-weighted Class Activation Mapping (Grad-CAM) (2017), enhancing the network prediction transparency. Kinger and Kulkarni [14] discussed several approaches and aimed to make deep models more interpretable in the context of plant disease recognition. The accuracy obtained using the VGG16 architecture was 98.15%. Grad-CAM was utilized to provide visual explanations that are interpretable to humans for the decisions made by the model.

Khattak et al. [15] proposed a CNN-based approach to tackle the diseases of citrus fruits. The model distinguished between healthy and unhealthy citrus fruits and leaves with an accuracy of 94.55%. The identification of the disease in citrus fruit increased the accuracy to 95.65%, thus proving the model to be a reliable tool for managing diseases among farmers. Nahiduzzaman et al. [16] proposed a CNN-XAI architecture for the classification of mulberry leaf diseases. A lightweight CNN framework was employed, which achieved remarkable accuracies of $95.05 \pm 2.86\%$ for three-class classification and $96.06 \pm 3.01\%$ for binary classification. In comparison to the traditional deep transfer learning models, the proposed model achieved better accuracy with a reduced number of parameters, layers, and computational complexity than those models. Additionally, SHAP brought more transparency to the model.

Arsenovic et al. [17] addressed a crucial problem by applying deep learning for precise classification of plant diseases. Traditional augmentation techniques and generative adversarial networks were used to build a huge dataset comprising 79,265 images of leaves. Experimental results showed that the model was able to detect plant diseases with an accuracy of 93.67% under varied conditions. Singh et al. [18] applied computer vision to reduce the agricultural loss in India caused by diseases of plants. Hand-labeling 2,598 samples from 13 plant species and 17 disease categories required 300 hours. The validation of the dataset was done by training three classification models, which achieved 31% improvement over the existing datasets. Many studies have used deep learning models to detect and diagnose plant diseases, comparing multiple architectures and visualization techniques. However, these studies have also highlighted vulnerabilities in the research, particularly the reliance on small datasets that do not reflect the diversity found in nature.

2 Methodology

Figure 1 illustrates the proposed methodology for identification and diagnosis of diseases affecting the leaves of several crops. After images of diseased plant leaves were captured and categorized, the dataset was processed to eliminate noise, convert the images to grayscale, enhance image quality, and resize them. Data augmentation was used to improve the dataset by introducing new image samples through rotation, translation, and random variations. The augmented images improved both the dataset and the model's performance. The model was trained with processed images in the next step. Then the model predicted diseases in new images, resulting in accurate diagnosis and classification of plant diseases.
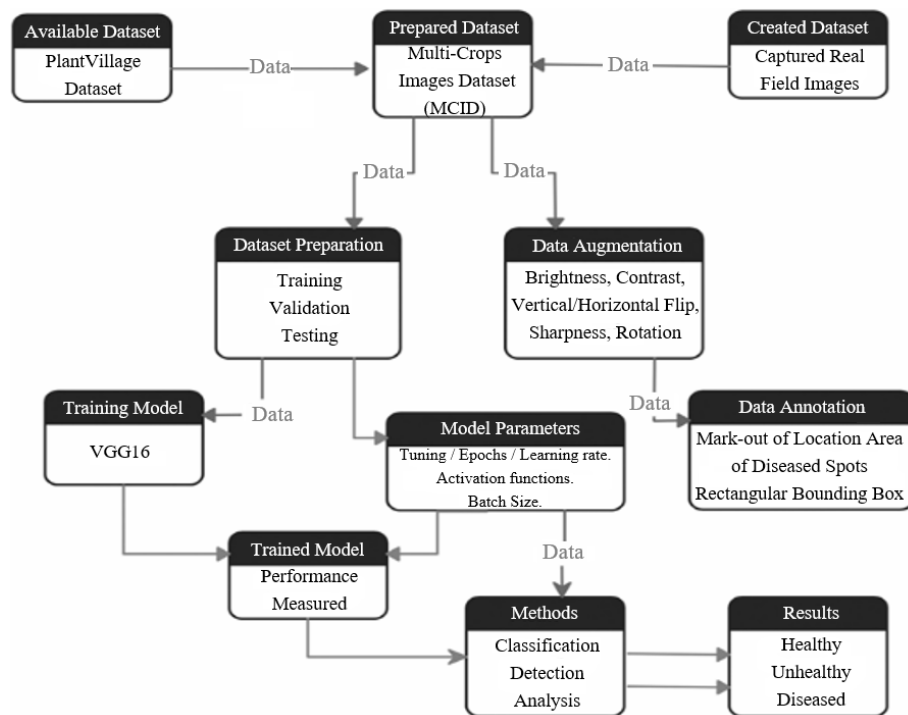

Figure 1. Flow chart of the proposed methodology

Figure 2 shows the field-collected healthy and diseased tomato leaves. The disease symptoms differ in pattern, site, and color in each of the images. The diseases of the tomato leaves, as shown by their symptoms, include grey spots, leaf mold and bacterial wilt [19]. Figure 3 shows photos of grapevine leaves taken in Nashik, Maharashtra, India. The dataset consists of leaf blight (1,076 images), healthy leaves (423 images), black measles (1,383 images), and black rot (1,180 images). Adjustments were made for categories A to D in terms of brightness and color for dataset enhancement. The second batch of tomato samples included the following diseases: tomato yellow leaf curl (3,209 cases), early blight (1,000 cases), leaf mold (952 cases), spider mite infestation (1,676 cases), bacterial spot (2,127 cases), Septoria leaf spot (1,771 cases), late blight (1,909 cases), unspecified spot disease (1,404 cases), and mosaic virus (373 cases). Deep learning greatly improved computer vision, particularly in image recognition and categorization. The generally proposed AI-based agricultural approaches for identifying and classifying leaf diseases in crops using CNNs involve stages of data acquisition, preprocessing, image segmentation, classification, and feature extraction. Feature extraction, image processing and classification were performed on the Google Colaboratory platform.
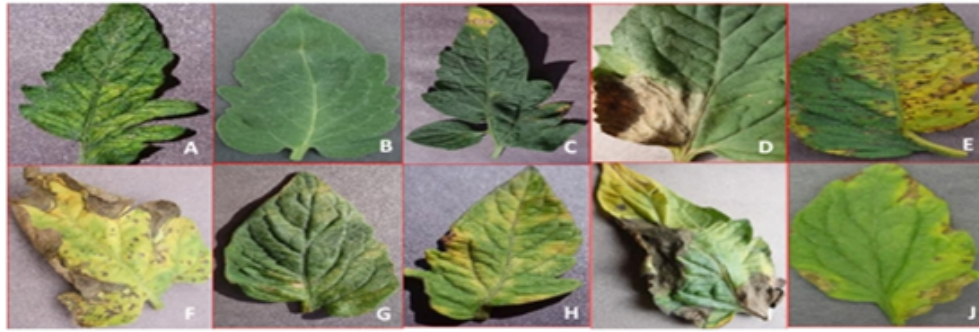
Figure 2. Tomato leaf sample images depicting: A) Mosaic Virus, B) Healthy Leaf, C) Target Spot, D) Late Blight, E) Bacterial Spot, F) Septoria Spot, G) Spider Mite, H) Leaf Mold, I) Early Blight, and J) Yellow Leaf

Figure 3. Grape leaf sample images depicting: A) Black Rot, B) Esca, C) Healthy, D) Leaf Blight (Isariopsis Leaf Spot)

2.1 Image Augmentation

Larger datasets enhance the efficiency of learning algorithms and also reduce overfitting. It is difficult and time-consuming to obtain real-time training datasets. Data augmentation introduces variety in training data for deep learning models. Some of the augmentation techniques include flipping, cropping, rotation, color changes, color augmentation based on Principal Component Analysis (PCA), noise reduction, Generative Adversarial Network (GAN), and Neural Style Transfer (NST) [20]. The Faster Dual-Region Integrated Attention Convolutional Neural Network (Faster DR-IACNN) model is a deep learning-based framework designed for the rapid and accurate identification of disease spots on grape leaves. From the original set of images, 4,449 images were used, and 62,286 additional images were generated through data augmentation methods.

Feature extraction of images was conducted during segmentation when fixed-length feature vectors were formed. The system was used to evaluate the color, texture, and shape of the images. Color properties were obtained from the Hue, Saturation, and Value (HSV) and Red, Green, and Blue (RGB) color spaces using methods such as means, confidence intervals, and smoothness. Texture features were captured from color images using the gray-level co-occurrence matrix (GLCM). This approach is essential for detecting diseases in plants.

2.2 Transfer Learning

Deep learning models require time, computational resources, and especially advanced GPUs along with massive training data to train and tune the models. However, these challenges can be easily resolved through transfer learning. Transfer learning in deep learning uses a pre-trained CNN for one task to exploit its knowledge for other tasks [21].

A multi-crop image dataset of 224 × 224 was used. The ResNet architecture was modified to accommodate the dataset. In almost all the topologies of ResNet, the layer preceding the softmax activation is a 7 × 7 average pooling layer. Smaller sizes of pooling allow the network to process smaller images. Transfer learning requires that the images be preprocessed to fit the models of multi-crop datasets.

3 Results and Discussion

3.1 CNN

CNNs contain convolutional, pooling, fully connected, and dense layers, as shown in Figure 4. Below is a detailed description of each layer.

Convolutional layers are primarily used to obtain features from the input images. The effectiveness of feature extraction is enhanced by applying these layers repeatedly [22]. The process of CNN feature extraction through

several layers is given by the following equation:

$$H_i = \varphi\left(H_{i-1}W_i + b_i\right) \tag{1}$$

where, H_i is the feature map, W_i is the weight, b_i is the offset, and φ is the ReLU.
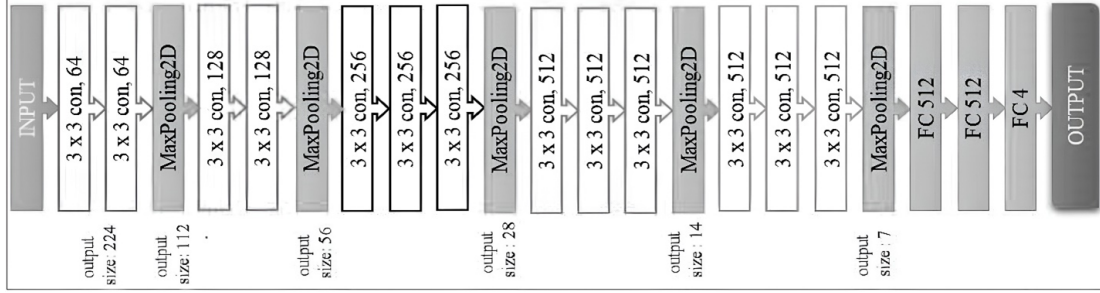

Figure 4. Schematic of the proposed CNN architecture

By combining feature information, pooling layers in CNNs eliminate the dimensionality of the feature map and help to optimize computation efficiency in picture processing. Max and average pooling are principal forms of these layers. In max pooling, the maximum value is selected from an image region, whereas in average pooling its mean is computed. Dropout layers are a form of regularization in training models, which helps improve the performance of the model. This reduces the dependence on particular neurons and thus prevents overfitting. A scaling factor makes this approach systematic for all activation functions [23]. The flatten layer reduces the pooled feature maps while preserving the channel information. It reshapes the data into a one-dimensional vector for the fully connected, dense layers.

A particular role of the fully connected layers is to help to classify the features extracted in the image. The softmax function predicts the properties from the previous layers and outputs multiclass classification by activating the output layers. In two-class classification problems, Multilayer Perceptron (MLP) models serve as classifiers within the layers of neural networks. Nonlinearity is introduced by the ReLU activation function in fully connected vectors. This architecture enables the implementation of complex decision boundaries. Some of the basic principles of SVMs are as follows:

$$\text{Minimize } \overset{n}{1/2} \sum_{j=1}^{N} W^2{}_1 1 + C \sum_{j=1}^{\xi j} \tag{2}$$

where, C serves as the tuning parameter, subject to the constraint $y_{i'}\left(\bar{W} \cdot \bar{X} + b\right) \cdot \geq 1 - \zeta_j \cdot (j = 1, 2, 3, \ldots, N)$.

Throughout the training and testing phases of the classification algorithm, the softmax parameter was set to $\gamma = 1$ and $C = 1$. The depth characterizes the ConvNet architecture. It deepens the network with more convolutional layers and hence improves the architecture. The accuracy in the recognition task is enhanced with the use of small 3×3 convolutional filters in all layers. These refined ConvNet models thus succeed in achieving state-of-the-art classification on datasets and localization, performing quite well across image recognition datasets even in very simple processing pipelines [24]. The ConvNets were trained on 224×224 RGB images. Pre-processing was limited to mean RGB value subtraction, calculated from the training dataset. The image went through convolutional layers and filters with a 3×3 receptive field. In one configuration, 1×1 convolutional filters were used to perform a linear transformation on the input channels followed by a non-linear function. To preserve spatial resolution, 3×3 filters require a stride and padding of 1 pixel each. Spatial pooling was performed by five max-pooling layers, each following specific convolutional layers. Not every convolutional layer was immediately shadowed by max pooling. A 2×2 pixel window with a stride of 2 was used to reduce the spatial dimensions while preserving salient features.

3.2 VGG16

A pre-trained VGG16 CNN model was used to classify healthy and unhealthy crop images to enhance the classification accuracy. The pre-trained VGG16 network helped the model determine the conditions of crop leaves. Besides, the CNN model also learned to detect and classify plant diseases from photographs taken in new fields [25].

The upgraded VGG model was implemented in configurations containing either 11 or 5 convolutional layers, each employing a uniform filter size of 3×3. The input image size was secured at 224×224. The images were pre-processed before being passed through the 3×3 convolutional layer. This layer applied a linear transformation

to the input channel using a 1 × 1 filter. Max pooling was performed using a 2 × 2 filter with a stride of 2. Each fully connected layer consisted of 4,096 units, maintaining consistent dimensionality across layers.

3.3 Performance Evaluation

The F1-score, Receiver Operating Characteristic (ROC) curve, accuracy matrix, and Area Under the Curve (AUC) were applied for segmentation performance assessment. Evaluation metrics also define the effectiveness of classifiers.
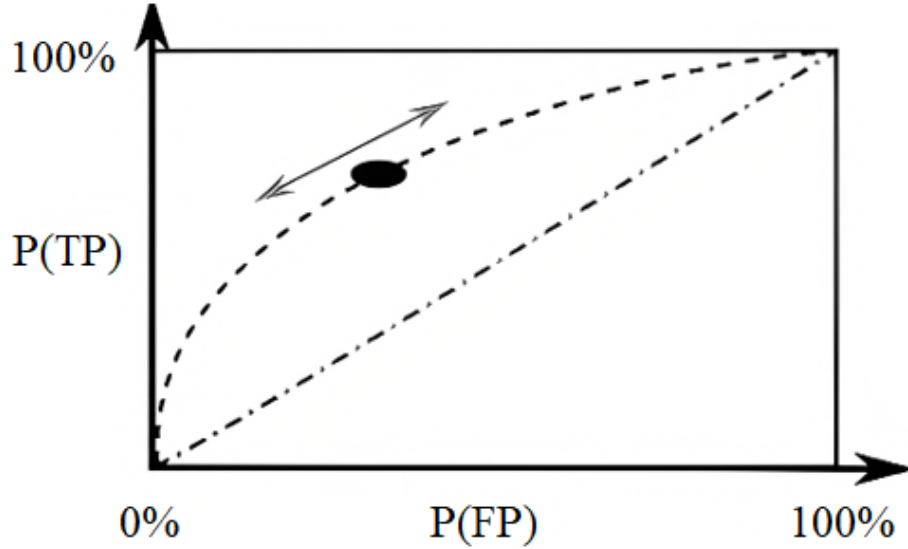

Figure 5. ROC curve: False positive rate vs. true positive rate

An overall assessment of the performance of the model on each class was conducted. Accuracy was calculated by dividing the number of correct predictions by the total number of predictions made. For a complete assessment, recall, F1-score and precision were also calculated. A mathematical representation of accuracy is given below.

$$AC = \frac{(TP + TN)}{(TP + FP + FN + TN)} \qquad (3)$$

where, *TP* is the true positive, indicating the correctly identified positive cases; *TN* is the true negative, indicating the correctly identified negative cases; *FP* is the false positive, indicating the incorrectly classified negative cases as positive; and *FN* is the false negative, indicating the incorrectly classified positive cases as negative.

Following are the classifier performance measures using evaluation metrics:

$$\text{TPR(Sensitivity)} = \frac{TP}{TP + FN}$$
$$\text{TNR(Specificity)} = \frac{TN}{TN + FP} \qquad (4)$$

$$\text{FPR} = \frac{FP}{(FP + TN)} \qquad (5)$$

where, *TPR* is the true positive rate, *TNR* is the true negative rate, and *FPR* is the false positive rate.

$$\text{Precision} = \frac{TP}{(TP + FP)}, \quad \text{Recall} = \frac{TP}{(TN + FN)} \qquad (6)$$

$$G - Mean = \left(\prod_{K=1}^{m} \text{Recall}_k \right) 1/m \qquad (7)$$

where, m is the total number of categories, and G denotes the accuracy ratio of the true negative rate to the false positive rate.

The mean average precision (mAP) of the algorithm measures precision, recall, and mean. mAP is used to evaluate image processing tasks and detection tasks. In terms of results, accuracy assesses the ratio of appropriately classified examples, while recall measures the proportion of correctly identified instances to the total relevant cases. The F1-score is yet another crucial performance measure, since it provides a balance between precision and recall which can be said to be more informative. It can be calculated as follows:

$$\text{F1 Score} \ = \ \frac{2 \times \ \text{Precision} \ \times \ \text{Recall}}{(\ \text{Precision} \ + \ \text{Recall}\ )} \tag{8}$$

The ROC curve is useful in assessing classification performance and addressing issues related to computational modeling. Figure 5 represents the connection between the false positive rate and the true positive rate at different thresholds.

3.4 Experimental Setup and Results

The model with the highest true negative rate detected the bad cases, whereas the model with the highest true positive rate classified the healthy cases. To reduce both training and testing times, an overall assessment was made using the Matthews Correlation Coefficient (MCC). MCC classifies only tough datasets reliably. Unlike accuracy, which can be misleading in imbalanced datasets, MCC takes into account all classification results, i.e., true negatives, true positives, false negatives, and false positives. Its values range between -1 (lowest classification) and +1 (perfect classification), and 0 denotes random predictions. The combination of hidden layers, number of epochs and hidden nodes, dropout rate, activation functions, learning rate and batch size impacts model optimization. Hyperparameter tuning – changing epochs, learning rate, hidden layers and activation functions in a systematic way – increases efficiency and performance. The model was adjusted to improve its accuracy and reduce the average loss.

Table 1. Comprehensive parameter settings for the trained model

Hyperparameter	Value/Setting
Crops	Grapes & Tomatoes
Image size	224 × 224 × 3
Convolutional layers	13
Max pooling layers	5
Activation functions	ReLU, Softmax
Dropout rate	0.15 / 0.25 / 0.50
Learning rate	0.00001 / 0.0001
Epochs	20 / 25 / 30 / 40 / 45

Table 2. Test results of classification

No. of Epochs	Learning Rate	Dropout Rate	No. of Images	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
40	0.00001	0.25	450	0.08	0.98	0.04	0.98
30	0.0001	0.50	450	0.11	0.95	0.05	0.98
45	0.00001	0.25	400	0.09	0.97	0.05	0.98
45	0.0001	0.25	750	0.08	0.96	0.05	0.98
30	0.0001	0.50	450	0.13	0.95	0.06	0.98
40	0.001	0.30	600	0.11	0.96	0.05	0.98

An experimental analysis was conducted on Google Colaboratory using the research tools developed by Google. This environment was equipped with Python programming and several pre-installed research libraries. Python 3, running on Google Compute Engine GPUs with 12.72 GB of RAM and 68.40 GB of disk space, was utilized for the experiment. The dataset was accessed by mounting Google Drive, and the platform's robust computational resources were utilized to train the model. This setup facilitated the execution of a Python program that converted images into arrays and retrieved them from the designated directory. All label images were binarized using Scikit-learn's label binarization function and retrieved from a designated folder for processing. The train-test-split function in Python

was used to split the dataset as training and testing datasets. The model parameters were set as shown in Table 1. The deep learning CNN was optimized using Adam, thereby overcoming sparse gradient noise.

The input network processed batches of 224 × 224 images containing 30 grapes and 25 tomatoes. The model was tested over epochs by varying the batch size and the learning rate. A max-pooling operation of 2 × 2 with ReLU activation was applied after each layer. In the final layer, multi-crop predictions were made using softmax activation. These training hyperparameters were tuned for performance as well.

The model attained an average accuracy of 98.40% and 95.71% for grapes and tomatoes, respectively. Learning rates were optimized through validation on a large multi-crop image dataset to enhance performance metrics. Accuracy increased with batch size and epoch configuration.

Table 3. Experimental results of the grape model for setting different parameters

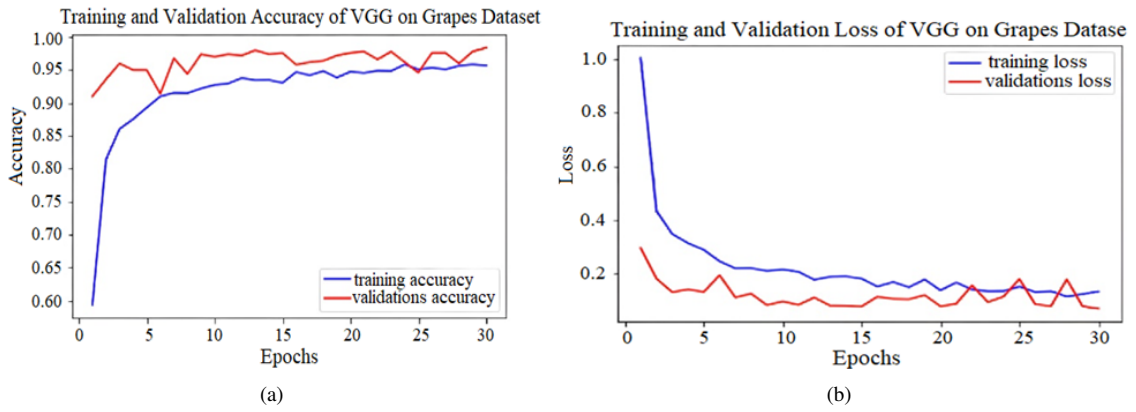| No. of Epochs	Learning Rate	Dropout Rate	No. of Images	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
22	0.0001	0.25	200	0.16	0.95	0.26	0.94
35	0.00001	0.20	180	0.22	0.92	0.31	0.90
30	0.00001	0.15	200	0.26	0.90	0.33	0.89
30	0.00001	0.25	200	0.30	0.88	0.37	0.88
30	0.00001	0.25	180	0.48	0.83	0.41	0.86
30	0.00001	0.50	200	0.52	0.82	0.45	0.85

(a) (b)

Figure 6. Training and validation performance: (a) accuracy and (b) loss curves for the VGG16 model on the grape dataset
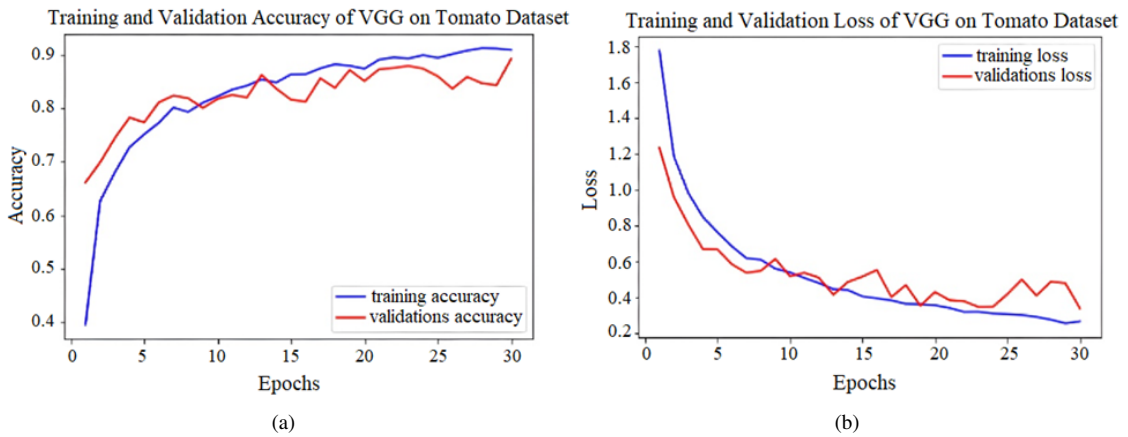

(a) (b)

Figure 7. Training and validation performance: (a) accuracy and (b) loss curves of the VGG16 model on the tomato dataset
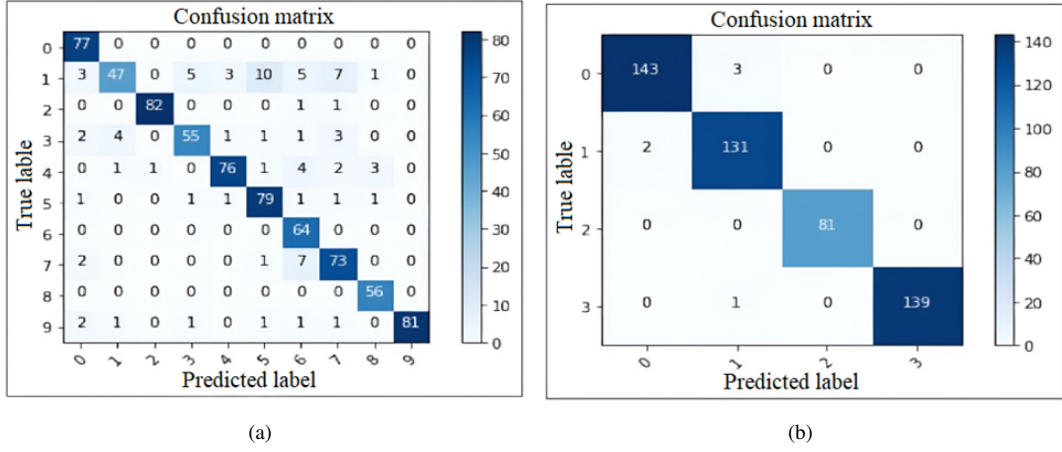
Figure 8. Confusion matrix results: (a) tomato dataset and (b) grape dataset
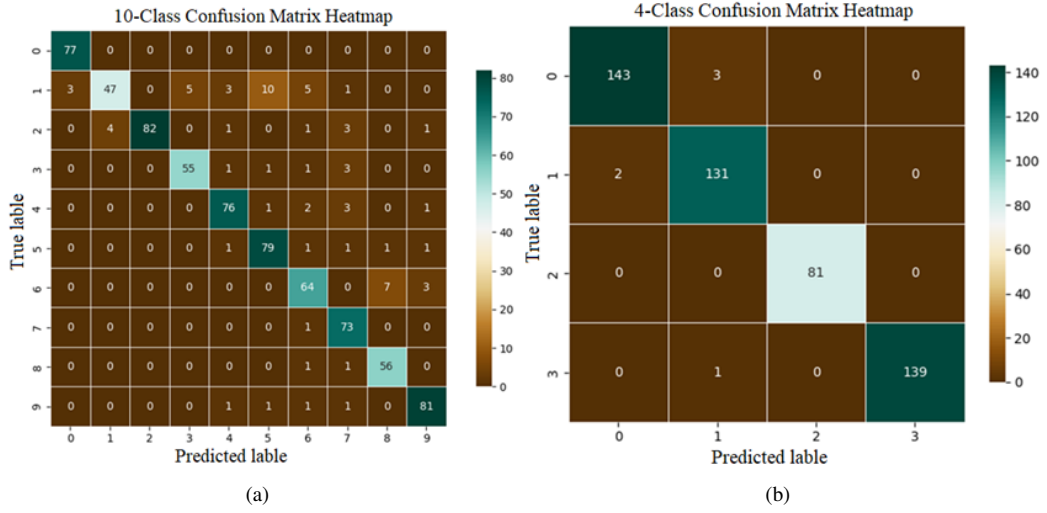

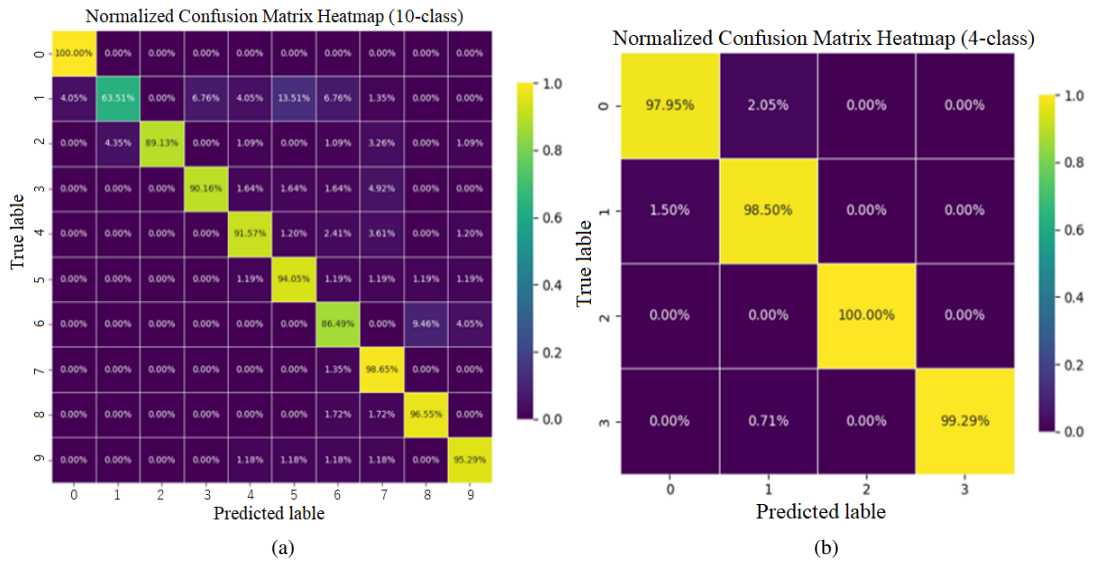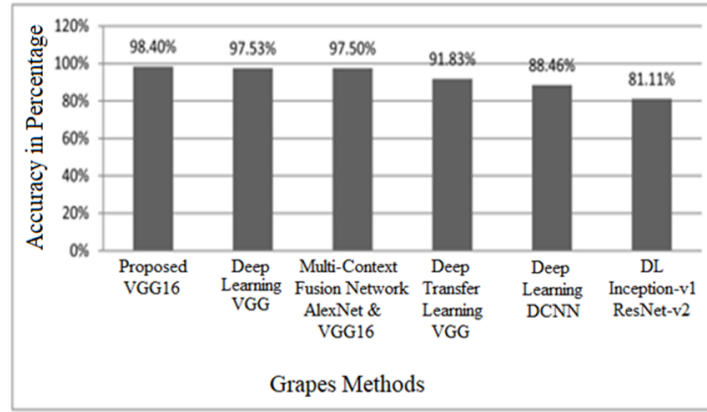**Figure 9.** Confusion matrix heat map: (a) tomato dataset and (b) grape dataset
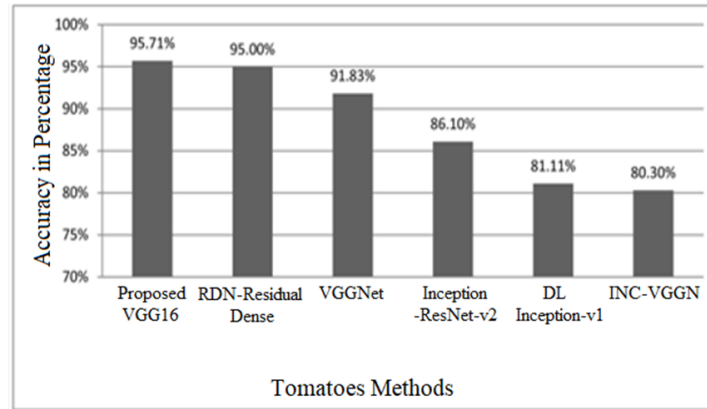

Figure 10. Normalized confusion matrix heat map: (a) tomato dataset and (b) grape dataset

(a)

(b)

Figure 11. Comparison of different models based on accuracy with the proposed VGG16 model for (a) the grape dataset and (b) the tomato dataset

Crop leaf images were employed to train the model to identify and classify disease types using transfer learning techniques, including VGG16. The dataset was distributed into 80:10:10 ratio for training, validation, and testing. Changes in model training parameters were evaluated by monitoring training and validation accuracy. These experiments were conducted on Google Colaboratory, equipped with 12.50 GB of RAM. In these experiments, the learning rate, epochs, dropout rate and the number of images were set to different values. These parameters influence accuracy, training loss, validation loss, and accuracy. The grape and tomato leaf samples were used to test the model. The experiments for the grape and tomato datasets are described in Table 2 and Table 3. The model was trained, tested, and validated on the leaves of grapes and tomatoes. Figure 6 and Figure 7 reflect the training along with validation accuracy and loss for both types of plants. Figure 8 shows the confusion matrix for classifying grape and tomato leaves. Figure 9 and Figure 10 show the confusion matrix heat maps and normalized confusion matrix heat maps for the tomato and grape datasets. Figure 11 presents a comparison of different models based on accuracy with the proposed VGG16 model for the grape and tomato datasets, respectively. The proposed process was assessed on a real-field image dataset with diverse backgrounds and lighting variations.

Data augmentation techniques such as random rotation, flipping, and scaling were used along with pre-processing to make the sample images more diverse and combat overfitting. These augmented the training dataset and thus strengthened the model. These processes are described as follows:

a) Image resizing: All images were resized to 224 × 224 pixels to be compatible with the model. Diversity was increased by data augmentation of at least 200 healthy and sick images.

b) Image preprocessing: This kept proportional ratios and yet maintained the structural information. The method gave more clarity to the image while reducing distortion.

c) Dataset partitioning and training: Random selection of images was used.

d) Validation and testing: The model was tested on both old and new images.

Results acquired were paralleled with the actual classes and hence the model's performance was evaluated

on control effectiveness. A modified Residual Dense Network (RDN) model used sets of residual blocks along with a Densely Connected Convolutional Network (DenseNet) to detect diseases in tomato leaves. After image normalization and convolutional residual modules, the dense layer achieved 95% accuracy in classifying tomato disease images on the disease dataset [26]. The Inception-ResNet-v2 model with the ReLU activation function obtained an accuracy of 86.1% in the AI Challenger Competition 2018 [27]. Under cluttered background conditions, the VGGNet model scored an accuracy of 91.83%. Using INC-VGGN, "Phaeosphaeria spot" and "maize eyespot" diseases were detected with an accuracy of 80.38% [27] (Figure 11).

4 Conclusion

A dataset comprising diseased leaves from two crop types was collected and prepared for this research. The VGG16 model, based on CNNs, was used for data augmentation, dataset preprocessing, training, and testing. The results were improved by designing the model and testing it against already available datasets and methods. The classification accuracy was 98.40% for grape diseases and 95.71% for tomato diseases. Identification of diseased leaves in field crops must be under agri-research development. The proposed system outperformed under all conditions and thus can contribute to advancements in agricultural practices. The purpose of this research is to enhance the agricultural sector and food security through high-quality production. Future work will involve gathering high-quality datasets with deep learning applications on images of crop leaves: more detailed analysis of diseases in crops using CNN models based on Inception V3 and ResNet principles. This research will empower the farmers by increasing their income, resulting in an increase in the national GDP.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M. Braik, and Z. Al-Rahamneh, "Fast and accurate detection and classification of plant diseases," *Int. J. Comput. Appl.*, vol. 17, no. 1, pp. 31–38, 2011. https://doi.org/10.5120/2183-2754

[2] D. Al Bashish, M. Braik, and S. Bani-Ahmad, "Detection and classification of leaf diseases using k-means-based segmentation and neural-networks-based classification," *Inf. Technol. J.*, vol. 10, no. 2, pp. 267–275, 2011. https://doi.org/10.3923/itj.2011.267.275

[3] J. G. Arnal Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," *SpringerPlus*, vol. 2, no. 1, p. 660, 2013. https://doi.org/10.1186/2193-1801-2-660

[4] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agric.*, vol. 147, pp. 70–90, 2018. https://doi.org/10.1016/j.compag.2018.02.016

[5] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Front. Plant Sci.*, vol. 7, p. 1419, 2016. https://doi.org/10.3389/fpls.2016.01419

[6] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Comput. Electron. Agric.*, vol. 154, pp. 18–24, 2018. https://doi.org/10.1016/j.compag.2018.08.048

[7] R. Bensoltane and T. Zaki, "Towards arabic aspect-based sentiment analysis: A transfer learning-based approach," *Soc. Netw. Anal. Min.*, vol. 12, no. 1, p. 7, 2022. https://doi.org/10.1007/s13278-021-00794-4

[8] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agric.*, vol. 145, pp. 311–318, 2018. https://doi.org/10.1016/j.compag.2018.01.009

[9] M. H. K. Mehedi, A. K. M. S. Hosain, S. Ahmed, S. T. Promita, R. K. Muna, M. Hasan, and M. T. Reza, "Plant leaf disease detection using transfer learning and explainable ai," in *2022 IEEE 13th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2022, pp. 166–170. https://doi.org/10.1109/IEMCON56893.2022.9946513

[10] S. Ramesh, R. Hebbar, M. Niveditha, R. Pooja, N. Shashank, and P. V. Vinod, "Plant disease detection using machine learning," in *2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C)*, 2018, pp. 41–45. https://doi.org/10.1109/ICDI3C.2018.00017

[11] M. A. Jasim and J. M. Al-Tuwaijari, "Plant leaf diseases detection and classification using image processing and deep learning techniques," in *2020 International Conference on Computer Science and Software Engineering (CSASE)*, 2020, pp. 259–265. https://doi.org/10.1109/CSASE48920.2020.9142097

[12] S. S. Harakannanavar, J. M. Rudagi, V. I. Puranikmath, A. Siddiqua, and R. Pramodhini, "Plant leaf disease detection using computer vision and machine learning algorithms," *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 305–310, 2022. https://doi.org/10.1016/j.gltp.2022.03.016

[13] M. Benito Fernández, D. L. Martínez, A. González-Briones, P. Chamoso, and E. S. Corchado, "Evaluation of XAI models for interpretation of deep learning techniques' results in automated plant disease diagnosis," in *Sustainable Smart Cities and Territories International Conference*, Cham, 2023, pp. 417–428.

[14] S. Kinger and V. Kulkarni, "Explainable AI for deep learning-based disease detection," in *Proceedings of the 2021 Thirteenth International Conference on Contemporary Computing*, 2021, pp. 209–216. https://doi.org/10.1145/3474124.3474154

[15] A. Khattak, M. U. Asghar, U. Batool, M. Z. Asghar, H. Ullah, M. Al-Rakhami, and A. Gumaei, "Automatic detection of citrus fruit and leaves diseases using deep neural network model," *IEEE Access*, vol. 9, pp. 112 942–112 954, 2021. https://doi.org/10.1109/ACCESS.2021.3096895

[16] M. Nahiduzzaman, M. E. H. Chowdhury, A. Salam, E. Nahid, F. Ahmed, N. Al-Emadi, M. A. Ayari, A. Khandakar, and J. Haider, "Explainable deep learning model for automatic mulberry leaf disease classification," *Front. Plant Sci.*, vol. 14, p. 1175515, 2023. https://doi.org/10.3389/fpls.2023.1175515

[17] M. Arsenovic, M. Karanovic, S. Sladojevic, A. Anderla, and D. Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, vol. 11, no. 7, p. 939, 2019. https://doi.org/10.3390/sym11070939

[18] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, 2020, pp. 249–253. https://doi.org/10.1145/3371158.3371196

[19] A. S. Paymode, V. B. Malode, and U. B. Shinde, "Artificial intelligence in agriculture for leaf disease detection and prediction: A review," *Artif. Intell. Agric.*, vol. 13, pp. 3565–3573, 2020.

[20] J. A. Pandian, G. Geetharamani, and B. Annette, "Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques," in *2019 IEEE 9th International Conference on Advanced Computing (IACC)*, 2019, pp. 199–204. https://doi.org/10.1109/IACC48062.2019.8971580

[21] P. Nevavuori, N. Narra, and T. Lipping, "Crop yield prediction with deep convolutional neural networks," *Comput. Electron. Agric.*, vol. 163, p. 104859, 2019. https://doi.org/10.1016/j.compag.2019.104859

[22] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Comput. Electron. Agric.*, vol. 173, p. 105393, 2020. https://doi.org/10.1016/j.compag.2020.105393

[23] S. Y. Liu, "Artificial intelligence (ai) in agriculture," *IT Prof.*, vol. 22, no. 3, pp. 14–15, 2020. https://doi.org/10.1109/MITP.2020.2986121

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015- Conference Track Proceedings*, 2015. https://arxiv.org/abs/1409.1556

[25] M. Alencastre-Miranda, R. M. Johnson, and H. I. Krebs, "Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties," *IEEE Trans. Ind. Inform.*, vol. 17, no. 2, pp. 787–794, 2021. https://doi.org/10.1109/TII.2020.2992229

[26] C. Zhou, S. Zhou, J. Xing, and J. Song, "Tomato leaf disease identification by restructured deep residual dense network," *IEEE Access*, vol. 9, pp. 28 822–28 831, 2021. https://doi.org/10.1109/ACCESS.2021.3058947

[27] Y. Ai, C. Sun, J. Tie, and X. Cai, "Research on recognition model of crop diseases and insect pests based on deep learning in harsh environments," *IEEE Access*, vol. 8, pp. 171 686–171 693, 2020. https://doi.org/10.1109/ACCESS.2020.3025325