Journal of Animal & Plant Sciences, 35(2): 2025, Page: 381-389
ISSN (print): 1018-7081; ISSN (online): 2309-8694
https://doi.org/10.36899/JAPS.2025.2.0032

CNN-BASED DETECTION OF POWDERY MILDEW AND RUST IN APPLE ORCHARDS FOR OPTIMIZING CROP MANAGEMENT

A. A. Al-Zubi

Department of Computer Science, Community College, King Saud University, Riyadh, Saudi Arabia
Corresponding Author email: aalzubi@ksu.edu.sa
https://orcid.org/0000-0001-8477-8319

ABSTRACT

In many parts of India, apple trees are among the most popular crops. Large amounts of apples are exported annually, which has a major positive impact on the country's economy. However, a number of diseases in apple trees are common. They indicate a significant risk to apple production and lead to significant financial losses for producers. These diseases mostly affect the leaves of apple plants. In a country where a significant portion of the workforce is employed in agriculture, prompt identification and management of such diseases are essential. It used to take a lot of time and effort to diagnose diseases in apple plants via laboratory testing. Machine Learning (ML) methods offer a fast and accurate detection of diseased leaves in the apple orchard. This study aimed to develop a robust Convolutional Neural Network (CNN) model for identifying apple leaf diseases. A dataset comprising 1,532 images categorized into Healthy, Powdery mildew, and Rust classes was used. The CNN model consisted of six convolutional layers, six max-pooling layers, a flatten layer, and fully connected layers. Images were pre-processed (resized to 256x256 pixels, normalized, and augmented) to improve computational efficiency. The model was evaluated using metrics such as accuracy, precision, recall, F1-score, and a confusion matrix. The model achieved a training accuracy of 98.02%, validation accuracy of 85.17%, and overall accuracy of 91.34%. Precision and recall for individual classes ranged from 86.05% to 96.55%. F1-scores showed balanced performance across categories, with a weighted average of 92.54%. These results demonstrate the model's effectiveness in classifying leaf conditions and its potential to enhance disease management in apple orchards and similar crops.

Keywords: Machine Learning, Apple Orchard, Crop Management, Convolutional Neural Network, Evaluation Metrics

Published first online February 22, 2025 Published final April 28, 2025

INTRODUCTION

Apple cultivation is a key part of agriculture in many parts of India. It contributes significantly to the economy through large-scale exports. However, apple orchards often face diseases that reduce yield and cause financial losses for farmers. Early detection of these diseases is essential to protect crops and ensure productivity. Traditional methods for disease detection rely on laboratory techniques. These methods are slow and not practical for large orchards. Machine learning (ML) has emerged as a faster and more reliable solution. It helps identify diseased leaves in less time and allows farmers to act promptly to prevent losses.

Recent advancements in technology have revolutionized agriculture, particularly in disease detection (Apolo-Apolo *et al*., 2020; Chu *et al*., 2021; Jia *et al*., 2022). ML methods provide fast and reliable solutions for identifying plant diseases. These algorithms analyze large datasets, extract patterns, and make precise predictions. Among these, convolutional neural networks (CNNs) stand out as powerful tools for image-based tasks. Their ability to process complex visual features makes them highly effective in detecting diseases in plant leaves.

Based on ML features, Mahato *et al*. (2022) developed for Convolutional Neural Network (CNN) model for Apple plant leaf diseases using images. The model was trained on the PlantVillage dataset. It achieved the highest overall accuracy of 99.31% with low training time and testing time of 5.1 ms per image. Ji *et al*. (2022) developed a real-time apple detection system using ShufflenetV2-YOLOX. The ShufflenetV2 network was improved with a Convolutional Block Attention Module (CBAM) and an Adaptive Spatial Feature Fusion (ASFF) module. These enhancements significantly boosted detection accuracy. The model achieved an average precision (AP) of 96.76%, a precision of 95.62%, and a recall of 93.75%. It also recorded an F1 score of 0.95 and operated at a high detection speed of 65 frames per second. Ahmed and Yadav (2024) created a dataset of 10,000 annotated RGB images representing foliar diseases in Kashmiri apple plants. They developed a Faster R-CNN model to identify diseases in real time. The proposed framework achieved a 92% accuracy, demonstrating its potential to transform orchard management and support apple growers in the Kashmir Valley.

Kaur *et al*. (2024) explored the detection of Apple Mosaic Disease (AMD) using a combination of Convolutional Neural Networks (CNN) and Random Forest (RF) models. The study used a rigorously preprocessed dataset of healthy and diseased apple leaves to train the models. CNN achieved a high diagnosis accuracy of 97.08%, showcasing its ability to identify complex AMD patterns. Tamana and Sharma (2024) proposed an enhanced YOLOv8 model for detecting and classifying apple leaf diseases, including black rot, apple scab, and cedar apple rust. The model was fine-tuned using the PlantVillage dataset, optimizing detection speed and accuracy. Through end-to-end training, the model achieved an impressive accuracy of 99.5% and a recall rate of 97.43%. Sulaiman *et al*. (2023) proposed a fine-tuned EfficientNetB3 model for quick and accurate apple foliar disease detection. The dataset used consisted of 23,187 RGB images representing 11 different apple diseases. The performance of the EfficientNetB3 model was compared with four other transfer learning models: InceptionResNetV2, ResNet50, AlexNet, and VGG16. The EfficientNetB3 outperformed the other models and achieved an accuracy of 89%.

Many advanced ML methods have been introduced to detect plant diseases. Liu *et al*. (2022) proposed a real-time-based YOLOX-ASSANano DL network design for apple leaf disease detection. The experiments were carried out on the Multi-Scene Apple Leaf Disease dataset (MSALDD). It shows a 91.08% mean average accuracy. Maheswari *et al*. (2021) discussed the use of CNN-based semantic segmentation techniques in intelligent fruit yield estimation for orchards. Challenges in terms of data sampling, collection, annotation and augmentation difficulty were also presented. Tang *et al*. (2023) developed YOLO-Oleifera model. which is a fruit detection model for oil-seed camellia fruit in orchard environments. The model uses k-means++ clustering algorithm and convolutional kernels to determine bounding box priors for camellia fruit sizes. It uses bounding boxes for adaptive stereo matching and precise positioning. YOLO-Oleifera achieves an AP of 0.9207 with a compact model size of 29 MB and an average detection time of 31 ms per fruit image. Thakur *et al*. (2022) discussed the global issue of crop diseases and proposed vision-based ML techniques for pest and disease control. They reviewed 1337 articles, focusing on Chinese and Indian researchers. Zhang *et al*. (2022) developed a framework using UAV and ground-based RGB image data to assess flowering intensity in a Dutch Elstar apple orchard. The framework uses automated techniques for point cloud reconstruction and unsupervised ML methods to train two linear regression models. Both models show robust performance, with R2 values exceeding 0.65 and significant correlations between the image-derived flower index and field-counted flower cluster number. Zhang *et al*. (2022) proposed a hybrid YOLOv3 algorithm for detecting damaged apples, using Rao-1 clustering to optimize anchor box sizes. The method generated representative anchor boxes for both normal and damaged apple detection. The Combined calculation of YOLOv3 and Fast R-CNN algorithms shows better performance metrics than a single one. The achievement of maximum mean average accuracy was one positive aspect of the combined model. This intelligent Apple detection model for the classification of diseased and normal classes can be beneficial for farmers. The automated DL algorithm for identifying citrus diseases in orchards was proposed (Zhang *et al*. 2022). The detection network algorithm was used to identify complex backgrounds and categorize fruits. The algorithm testing was done on 1524 images. The outcomes showed an overall accuracy of 0.890 and an F1 score of 0.872. The capability of ML methods has been utilized in recent studies including several sectors such as animal, legume, health, and finance (Kim and AlZubi, 2024; Min *et al*., 2024; Wasik and Pattinson, 2024; Porwal *et al*., 2024).

In this work, a sequential-CNN architecture consisting of convolutional, max-pooling, dense, and dropout layers is designed to detect disease in apple orchards. The input images have dimensions of 256x256 pixels. The model contains five Conv2D layers, two MaxPooling2D layers, one Flatten layer, two Dense layers, and a Dropout layer. To prevent overfitting, a dropout rate of 0.5 is applied. The outcomes were evaluated in terms of confusion matrix, and overall accuracy with other performance matrices. The study can be used as a reference for future work to detect leaf diseases in apples or other trees at an early stage.

MATERIALS AND METHODS

Dataset: The dataset is acquired from the open-source Kaggle database. The dataset is divided into three categories: Healthy, Powdery, and Rust (Figure 1). which correspond to various plant conditions. It has 1532 images altogether and is separated into subgroups for training, testing, and validation (Table 1). The dataset was divided into 80:10:10 ratios for training, testing, and validation sets.

Data preprocessing techniques: The process of making CV and ML models from images is not easy because of issues with complexity, accuracy, and suitability (Charte *et al*., 2021). Before making any computational model, it is important to preprocess the data.

In the data preprocessing method, first, all images were resized to 256x256 pixels to standardize the input dimensions. This was necessary because the dataset included images from different cameras with varying resolutions. Next, normalization was applied by scaling the pixel values to a range of 0 to 1. This step ensures faster neural network convergence. Data augmentation techniques were used to increase dataset diversity and prevent overfitting. These included random rotations, horizontal flipping, and zooming. Augmentation was applied only to the training dataset, ensuring that the validation and test datasets remained representative.

Table 1. No. of image division in different datasets.

Data Type	Healthy	Powdery	Rust	Total
Training set	428	410	414	1252
Test set	50	50	50	150
Prediction set	50	50	50	150
Grand Total	**528**	**500**	**504**	**1532**

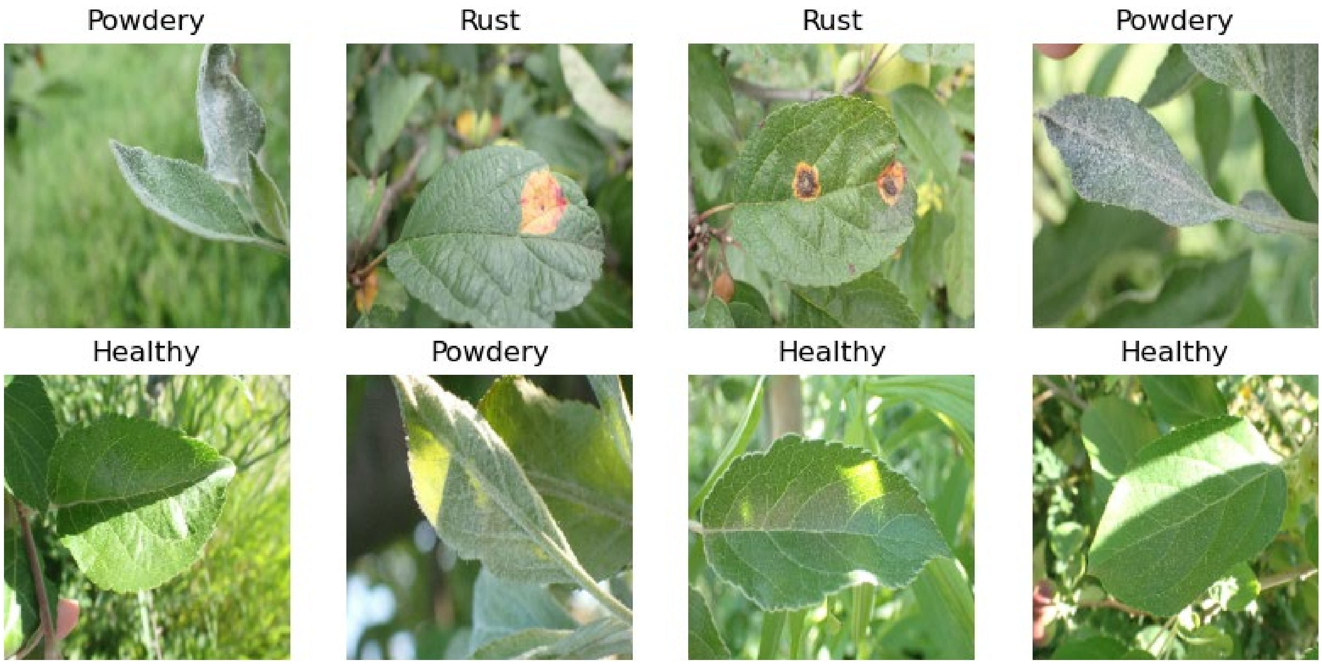

Figure 1. Healthy and diseased leaves on apple orchard.

Model Architecture: The architecture of a neural network for processing image data consists of many layers. Each layer performs certain functions and changes the dimension of input data. The Output Shape column displays the dimensions of the output data after it has been trained through each layer. The total number of parameters displays weights and biases in each layer, which determines the model's complexity. Convolutional layers (Conv2D) utilise filters to extract information from input images. The essential data is preserved in MaxPooling2D layers by reducing the feature maps' spatial dimensions. The flatten layer reshapes the output from the layers above into a one-dimensional vector. Dense layers are fully connected layers that employ extracted data to accomplish classification. The dropout layer randomly deactivates certain neurons during training to prevent overfitting. The summary includes the total number of parameters in the model, both trainable and non-trainable.

In this work, the sequential CNN model is employed to extract features from input images. The several layers in the computations are taken (Figure 2). The Conv2D layers apply convolution operations on the incoming data. The first Conv2D layer receives an input of size 256x256 and creates a feature map with dimensions 254x254. Filters are used to extract features from the input image during the convolution step. After working on the 254x254 feature map, the second Conv2D layer creates a 252x252 feature map to extract more features from the input. The MaxPooling2D layers downsample the feature maps after the convolutional layers. The first MaxPooling2D layer reduces the spatial dimensions of the 252x252 feature map to 63x63. More complex patterns in the data are captured by the third Conv2D layer, which takes the 63x63 feature map and creates an output with dimensions of 61x61. The 61x61 feature map is converted into a 59x59 output by the fourth Conv2D layer, which improves the feature extraction procedure. Lastly, a 57x57 feature map is produced by the fifth Conv2D layer after it processes the 59x59 feature map. subsequently, the spatial dimensions are further reduced from 57x57 to 14x14 by the second MaxPooling2D layer. The Flatten layer finally transforms the 14x14 feature map into a one-dimensional array of

9800 elements and gets ready for input into the fully connected layers for classification. The dense layers in the neural network model represent fully connected layers. The first dense layer has 120 neurons with a total of 1,176,120 parameters. Next, the dense_1 layer has 12,100 parameters and 100 neurons. Hence, the dense_2 layer consists of 50 neurons with 5,050 parameters. After the second dense layer, a dropout layer of 0.5 with 0 parameters is employed to randomly remove neurons to minimise overfitting during training. Three neurons and 153 parameters make up the final dense layer. The model has 1,753,823 parameters, each of which can be trained to minimize loss function during the training phase. It has no non-trainable parameters, ensuring every parameter aids in the learning process.

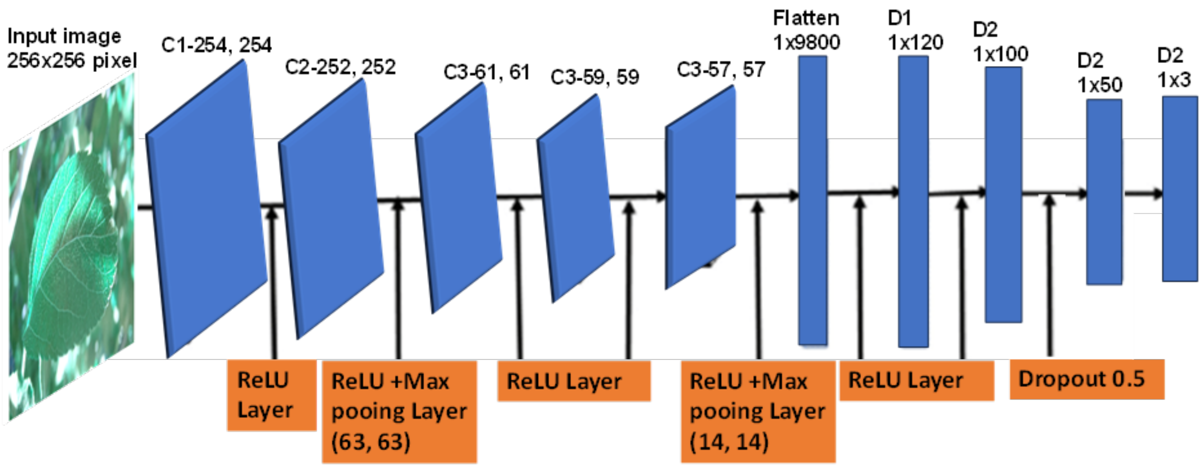

Figure 2. CNN model architecture

Table 1. Hyperparameters used in algorithm.

Hyperparameter	Value
Image Size	256 x 256
Batch Size	32
Channels	3 (RGB)
Number of Epochs	50
Optimizer	Adam
Learning Rate	0.0001
Loss Function	Sparse Categorical Crossentropy (from_logits=False)
Convolution Layers	6 layers
Pooling Layers	6 MaxPooling2D layers
Activation Functions	ReLU (hidden layers), Softmax (output layer)
Dropout Rate	0.5

Evaluation Metrics: There are four important factors that are used to determine the ability of a new model to work. these are the F1 score, the accuracy, the precision, and the recall. A confusion matrix is a presentation of a matrix that also provides the outcome of a computational method. Accuracy is the number of correct predictions out of all the cases. This is calculated as the sum of True Positive (TP) and True Negative (TN). The accuracy is $(TP + TN)/(TP + TN + FP + FN)$. While precision and recall are expressed as

$$Precision = TP/(TP + TN),$$
$$Recall = TP/(TP + FN)$$

The F1 score combines recall and precision. It demonstrates how well the models understand the need for accuracy and recall. The confusion matrix for healthy and diseased classes is shown below.

		Predicted Class		
		Healthy	Powdery mildew	Rust
Actual Class	Healthy	TP	FN	FN
	Powdery mildew	FP	TP	FN
	Rust	FP	FP	TP

RESULTS AND DISCUSSION

This section discusses the findings from the training and testing of a neural network model to detect conditions such as rust, powdery mildew, and normal leaves in an apple orchard. The percentage of properly identified samples relative to the total number of samples is known as accuracy. The training process over 50 epochs demonstrates the model's progressive improvement in accuracy and loss across both training and validation datasets (Figure 3). Initially, the model starts with a training accuracy of 41% and a high loss of 1.0872, while validation accuracy is 52.42% with a loss of 1.0167. By Epoch 5, training accuracy improves to 71.62%, and validation accuracy increases to 75.81%, accompanied by a significant reduction in loss values, indicating that the model is effectively identifying basic patterns.

During the next phase, from Epochs 6 to 15, the model shows rapid improvement, with training accuracy reaching 87.96% and validation accuracy climbing to 86.29% by Epoch 10. Loss values drop steadily, reflecting robust generalization. In the stabilization phase, from Epochs 16 to 30, training accuracy gradually rises above 95% by Epoch 20, and validation accuracy peaks at 95.97%. Occasional fluctuations in validation loss, such as 0.2479 in Epoch 28, highlight sensitivity to certain data patterns. In the final phase, from Epochs 31 to 50, the model plateaus in performance, achieving training accuracy close to 97% and maintaining validation accuracy between 95% and 97.7%.

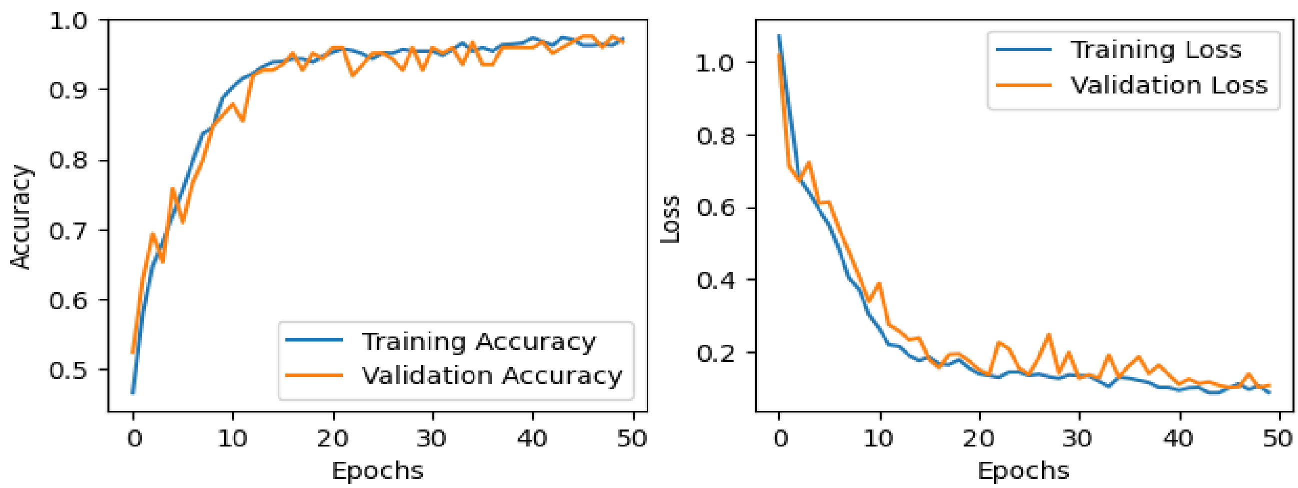

Figure 3. Model accuracy and loss graphs after 50 epochs.

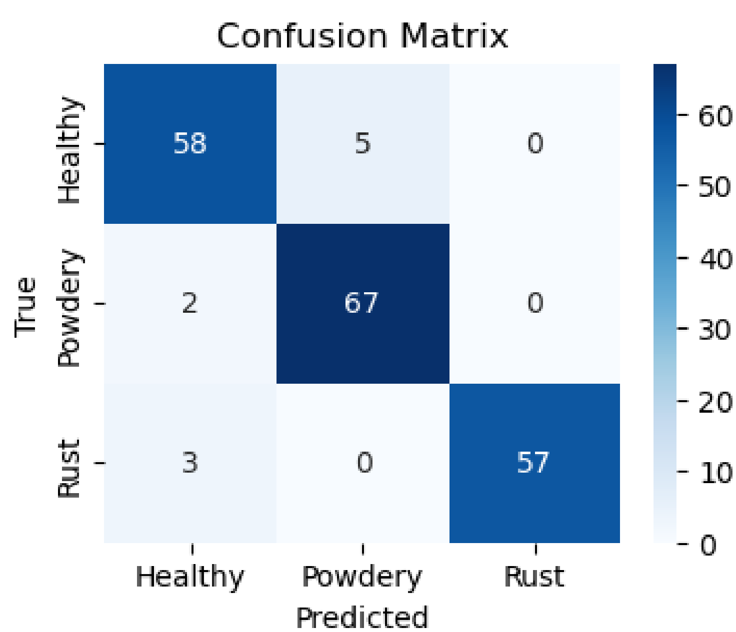

Figure 4. Confusion matrix

Validation loss minimizes to approximately 0.1014 in Epoch 46, indicating highly accurate predictions on unseen data. The overall training process demonstrates smooth convergence and effective generalization, supported by the alignment of training and validation metrics, with no signs of overfitting. The confusion matrix is an essential tool to evaluate the classification ability of the model (Figure 4). The predicted classes are shown in columns, whereas the actual classes are represented by rows. In every matrix cell, the number of instances in the real class matches the predicted class. The model demonstrated remarkable accuracy, with the majority of samples correctly classified. For Healthy samples, 58 were accurately identified, while 5 were misclassified as Powdery Mildew, and none were labeled as Rust. In the case of Powdery Mildew, the model successfully classified 67 samples, with only 2 misclassified as Healthy and none as Rust. Similarly, for Rust, 57 samples were correctly categorized, with 3 mistakenly identified as Healthy and no misclassifications as Powdery Mildew.

These results highlight the model's effectiveness in distinguishing between the three categories, with minimal misclassifications and no overlap between Rust and Powdery Mildew. The evaluation metrics are summarized for healthy, powdery mildew, and rust classes (Table 2). It measures the precision of positive predictions. The ratio of actual positive predictions to all expected positives is used to compute it.

Table 2. Classification metrics.

	precision	recall	F1-score	support
Healthy	0.9206	0.9206	0.9206	63
Powdery	0.9306	0.971	0.9504	69
Rust	1.000	0.95	0.9744	60
accuracy			0.9479	
macro avg	0.9504	0.9472	0.9484	192
weighted avg	0.949	0.9479	0.9481	192

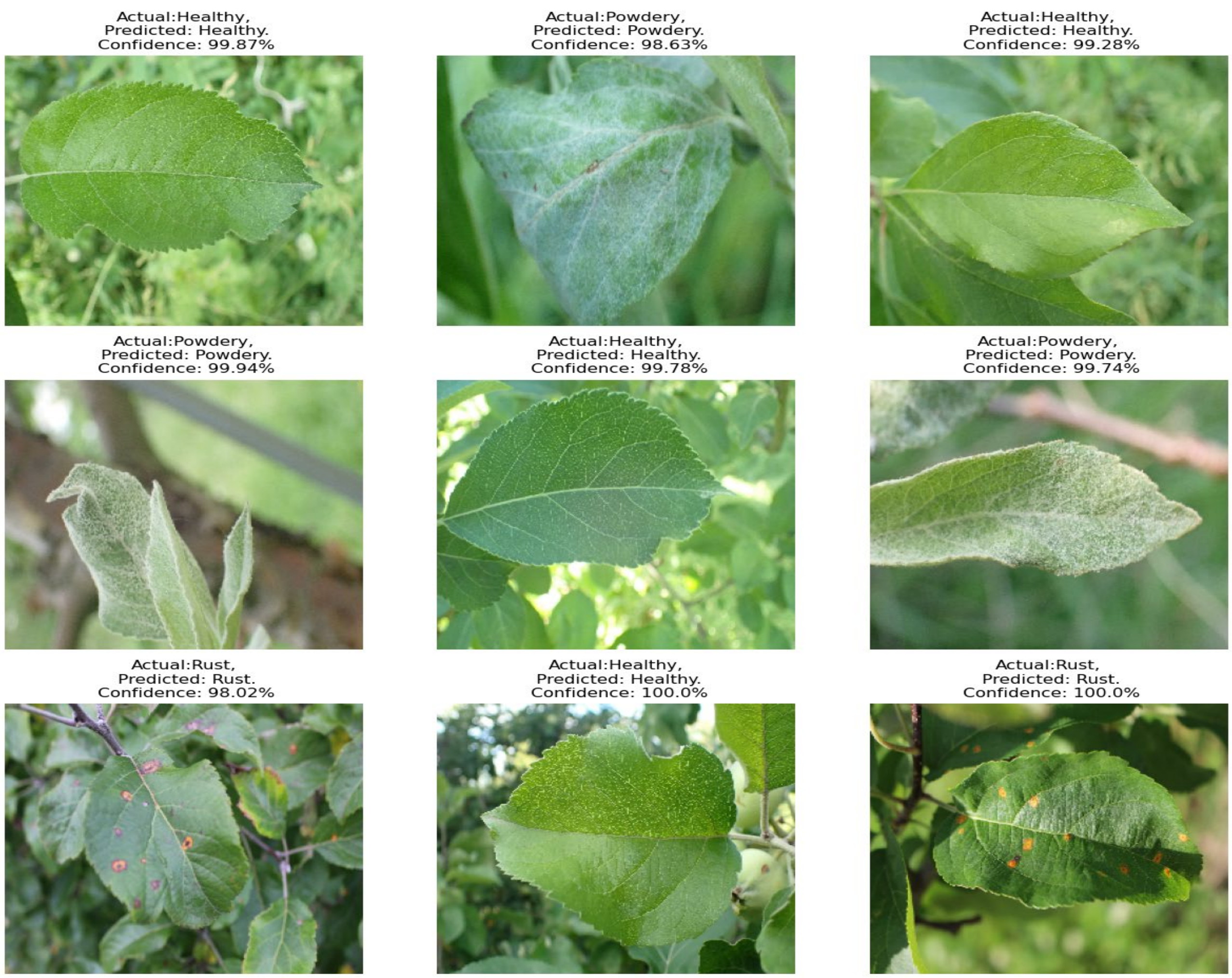

Figure 5. Prediction behavior for individual test samples with confidence score

Additionally, to assess the generalization ability of the trained CNN model on unseen data and analyze prediction behavior for individual test samples, images from the test dataset were passed through the trained model to predict their classes. The actual and predicted labels, along with confidence scores, were displayed in Figure 5. The performance of model was evaluated using ROC curves and AUC values for each class. Figure 6 shows the ROC (AUC) curves for all classes.

The Healthy class achieved an AUC of 0.7941. The Powdery Mildew class recorded the lowest AUC at 0.7688, indicating some challenges in differentiation. The Rust class had the highest AUC of 0.8220, showing strong classification accuracy. These results suggest the model performs well in identifying Rust. However, the lower AUC for Powdery Mildew highlights the need for further improvement.

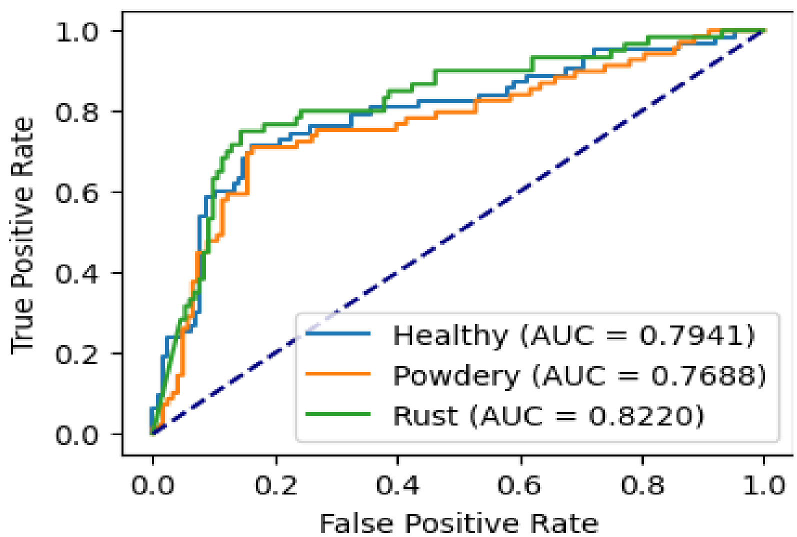

Figure 6. ROC(AUC) curve

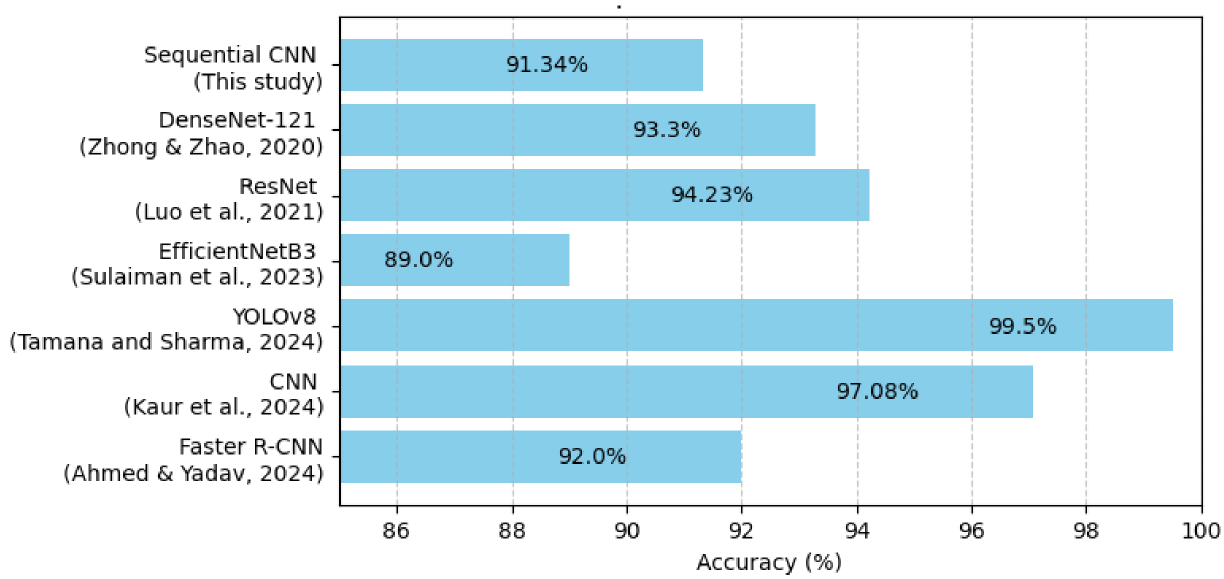

Figure 7. Comparison of models

In comparison to the other models, as shown in Figure 7, the Sequential CNN used in the presented study achieved an accuracy of 91.34%, which places it among the middle performers. The remarkable accuracy was achieved by the YOLOv8 model (99.50%) and CNN model (97.08%) (Tamana and Sharma, 2024; Kaur *et al.*, 2024). In contrast, the proposed CNN model shows a slightly lower accuracy than models such as ResNet (94.23%) and DenseNet-121 (93.30%) (Luo *et al.*, 2021; Zhong and Zhao, 2020), which both showed relatively strong results. However, CNN model does outperform EfficientNetB3 (89%) and Faster R-CNN (92%) (Sulaiman *et al*., 2023; Ahmed and Yadav, 2024), highlighting its competitive performance in comparison to these models.

Limitations and Future work: The proposed model demonstrated good performance on training and validation datasets, but several limitations need further attention. Its generalizability is uncertain, as it was tested on a controlled dataset. Variations in lighting and leaf conditions may affect performance in different orchards. Additionally, the model struggled with Powdery Mildew classification, shown by a lower AUC. This could be improved with more diverse samples or data augmentation. Future improvements include combining CNNs with transformer-based models like Vision Transformers (ViTs) to capture more complex data patterns. Adding interpretability methods, such as Grad-CAM, would enhance user trust. Additionally, creating lightweight versions of the model for mobile or edge devices would enable field-level disease detection.

Practical implications: Deploying the model in orchard management systems offers key benefits. It could enable real-time disease detection via drone or mobile apps, allowing timely intervention and reducing crop losses. However, challenges like latency, accuracy in the field, and implementation costs need to be addressed. The model's scalability is important, and expanding its dataset to include more diseases will ensure wider applicability. Automated disease identification could also reduce reliance on expert diagnostics, cutting costs and supporting sustainable farming. Collaboration with agricultural organizations could help scale deployment.

Conclusion: The research employs Convolutional Neural Network (CNN) architectures to increase the accuracy of apple plant disease classification. It creates and evaluates improved CNN-based models with a focus on two prevalent apple leaf diseases. The evaluation measures of the model can provide a complete understanding of its capabilities. The model's remarkable 91.34% overall accuracy indicates that all courses learn from its performance. Improved CNN-based models that provide automated disease identification allow for more effective disease management and preventative actions for apple orchards. It will be important for future studies to focus on enhancing and integrating these techniques in addition to classifying diseases other than apple leaf diseases. Further investigation is required to completely comprehend unified networks with enhanced categorization precision.

Acknowledgement: The authors would like to thank the editors and reviewers for their review and recommendations and also to extend their thanks to King Saud University for funding this work through the Researchers Supporting Project number (RSP2025R395), King Saud University, Riyadh, Saudi Arabia.

Funding Statement: This work was supported by the Researchers Supporting Project number (RSP2025R395), King Saud University, Riyadh, Saudi Arabia.

Author contributions: The author contributed toward data analysis, drafting and revising the paper and agreed to be responsible for all aspects of this work.

Data Availability Statement: The data analysed/generated in this study will be available from the corresponding authors upon reasonable request.

Declaration of Conflicts of Interests: The author declares that he has no conflict of interest.

Use of Artificial Intelligence: Not applicable

Use of Research Reporting Tool: Used Python for analysis

REFERENCES

Ahmed, I. and P.K. Yadav (2024). Predicting apple plant diseases in orchards using machine learning and deep learning algorithms. SN Computer Science, 5(6). https://doi.org/10.1007/s42979-024-02959-2

Apolo-Apolo, O.E., M. Pérez-Ruiz, J. Martínez-Guanter and J. Valente (2020). A cloud-based environment for generating yield estimation maps from apple orchards using UAV imagery and a deep learning technique. Frontiers in Plant Science, 11. https://doi.org/10.3389/fpls.2020.01086

Charte, D., F. Charte and F. Herrera (2021). Reducing data complexity using autoencoders with class-informed loss functions. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(12), 9549–9560. https://doi.org/10.1109/TPAMI.2021.3127698

Chu, P., Z. Li, K. Lammers, R. Lu and X. Liu (2021). Deep learning-based apple detection using a suppression mask R-CNN. Pattern Recognition Letters, 147, 206–211. https://doi.org/10.1016/j.patrec.2021.04.022

Dataset. Plant disease classification. Kaggle. https://www.kaggle.com/code/vad13irt/plant-disease-classification/input [Accessed On 23 May 2024]

Ji, W., Y. Pan, B. Xu and J. Wang (2022). A real-time apple target detection method for picking robots based on ShufflenetV2-YOLOX. Agriculture, 12(6), 856. https://doi.org/10.3390/agriculture12060856

Jia, W., Z. Wang, Z. Zhang, X. Yang, S. Hou and Y. Zheng (2022). A fast and efficient green apple object detection model based on Foveabox. J. King Saud University - Computer and Information Sciences, 34(8), 5156–5169. https://doi.org/10.1016/j.jksuci.2022.01.005

Kaur, A., V. Kukreja, P. Aggarwal, S. Thapliyal and R. Sharma (2024). Amplifying apple mosaic illness detection: combining CNN and random forest models. 2024 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), 1–5. https://doi.org/10.1109/ IATMSI 60426.2024.10503272

Kim, S. Y. and A. A. AlZubi (2024). Blockchain and artificial intelligence for ensuring the authenticity of organic legume products in supply chains. Legume Research. https://doi.org/ 10.18805/LRF-786

Liu, S., Y. Qiao, J. Li, H. Zhang, M. Zhang and M. Wang. (2022). An improved lightweight network for real-time detection of apple leaf diseases in natural scenes. Agronomy, 12(10), 2363. https://doi.org/ 10.3390/ agronomy 12102363

Luo, Y., J. Sun, J. Shen, X. Wu, L. Wang and W. Zhu (2021). Apple leaf disease recognition and sub-class categorization based on improved multi-scale feature fusion network. IEEE Access, 9, 95517–95527. https://doi.org/10.1109/ACCESS.2021.3094802

Mahato, D.K., A. Pundir and G.J. Saxena. (2022). An improved deep convolutional neural network for image-based apple plant leaf disease detection and identification. J. Institution of Engineers Series A, 103(4), 975–987. https://doi.org/10.1007/s40030-022-00668-8

Maheswari, P., P. Raja, O.E. Apolo-Apolo and M. Pérez-Ruiz. (2021). Intelligent fruit yield estimation for orchards using deep learning-based semantic segmentation techniques: a review. Frontiers in Plant Science, 12. https://doi.org/10.3389/fpls.2021.684328

Min, P.K., K. Mito and T.H. Kim. (2024). The evolving landscape of artificial intelligence applications in animal health. Indian J. Animal Research. https://doi.org/10.18805/IJAR.BF-1742

Porwal, S., M. Majid, S.C. Desai, J. Vaishnav and S. Alam. (2024). Recent advances, challenges in applying artificial intelligence and deep learning in the manufacturing industry. Pacific Business Review (International), 16(7), 143–152.

Sulaiman, A., V. Anand, S. Gupta, H. Alshahrani, M.S.A. Reshan, A. Rajab, A. Shaikh and A.T. Azar (2023). Sustainable apple disease management using an intelligent fine-tuned transfer learning-based model. Sustainability, 15(17), 13228. https://doi.org/10.3390/su151713228

Tamana and S. Sharma. (2024). Apple leaf disease prediction using modified YOLOv8 algorithm. 2024 International Conference on Integrated Circuits, Communication, and Computing Systems (ICIC3S), 1–6. https://doi.org/10.1109/ ICIC3S61846.2024.10603307

Tang, Y., H. Zhou, H. Wang and Y. Zhang (2023). Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. Expert Systems with Applications, 211, 118573. https://doi.org/10.1016/ j.eswa.2022.118573

Thakur, P. S., P. Khanna, T. Sheorey and A. Ojha (2022). Trends in vision-based machine learning techniques for plant disease identification: a systematic review. Expert Systems with Applications, 208, 118117. https://doi.org/ 10.1016/j.eswa.2022.118117

Wasik, S. and R. Pattinson (2024). Artificial intelligence applications in fish classification and taxonomy: advancing our understanding of aquatic biodiversity. FishTaxa, 31, 11–21.

Zhang, C., C. Mouton, J. Valente, L. Kooistra, R. Van Ooteghem, D. De Hoog, P. Van Dalfsen and P. F. De Jong (2022). Automatic flower cluster estimation in apple orchards using aerial and ground-based point clouds. Biosystems Engineering, 221, 164–180. https://doi.org/10.1016/j.biosystemseng.2022.05.004

Zhang, M., H. Liang, Z. Wang, L. Wang, C. Huang and X. Luo (2022). Damaged apple detection with a hybrid YOLOv3 algorithm. Information Processing in Agriculture. https://doi.org/10.1016/j.inpa.2022.12.001

Zhang, X., Y. Xun and Y. Chen (2022). Automated identification of citrus diseases in orchards using deep learning. Biosystems Engineering, 223, 249–258. https://doi.org/10.1016/j.biosystemseng.2022.09.006

Zhong, Y. and M. Zhao (2020). Research on deep learning in apple leaf disease recognition. Computers and Electronics in Agriculture, 168, 105146. https://doi.org/10.1016/j.compag.2019.105146.