

Research Article

Leaf-by-Leaf Diagnosis: A Custom CNN with Pyramidal Feature Extraction for Plant Disease Classification

Ahmad S. Lateef^{1, *, }

¹ College of Science, Mustansiriyah University, Baghdad, Iraq

ARTICLE INFO

Article History

Received 3 Apr 2026

Revised: 20 May 2026

Accepted 23 Jun 2026

Published 10 Jul 2026

Keywords

Plant disease,

Deep learning,

Convolutional Neural
Networks,

Image recognition,

Image processing.



ABSTRACT

Plant diseases threaten global agriculture, causing 20–40% yield losses and food insecurity. Current diagnostic methods are costly and lack scalability. While deep learning advances plant disease detection, there remains a need for CNNs with simpler architectures, better generalizability, and lower computational cost. This study presents a novel CNN for multi-class classification of 38 diseases. Trained on a public dataset of over 87,000 RGB images, the architecture comprises five convolutional blocks (filters 32–512) with max pooling and dropout (0.25, 0.4), followed by a 1,500-unit dense layer and SoftMax output. Optimized with Adam (lr=0.0001) and categorical cross-entropy, the model achieved 98% training and 96% validation accuracy with approximately 28.7 million parameters significantly fewer than transfer learning architectures. These results demonstrate an effective balance between predictive performance and computational efficiency, positioning the model as a promising tool for real-world agricultural deployment.

1. INTRODUCTION

It can be asserted that agriculture plays a crucial role in ensuring the food security and stability of the world economy through providing millions of people with food and jobs [1]. Contemporary systems of agriculture are characterized by numerous difficulties, and one of them is related to plant diseases that adversely affect the productivity of crops. According to empirical evidence, losses caused by pests and diseases make up around 20%–40% of global food production each year and result in economic damage worth more than \$310 billion USD [2]. Moreover, there is a forecast stating that in 2050, the world population will amount to 9.8 billion people and thus the agricultural production should increase by 60%. Consequently, the creation of an innovative approach to the identification of crop health and diseases becomes critical in order to avoid the spreading of pathogens, reduce chemical pesticide usage, and support the Sustainable Development Goals of zero hunger and sustainable agriculture [3].

Conventional methodologies adopted to diagnose plant diseases primarily involve visual inspection by agronomists or molecular analysis in laboratories. Although these methodologies prove to be quite effective, they are time-consuming, labor-intensive, expensive, and unfeasible for extensive use in practice [4]. Computer vision paired with artificial intelligence, specifically, deep learning, has recently proven to be an innovative methodology to overcome these drawbacks. The implementation of deep learning algorithms allows automatic, fast, non-destructive, and precise disease detection and classification using leaf image data without requiring any expert knowledge or expensive laboratory equipment [5]. Among all existing architectures, Convolutional Neural Networks (CNNs) have been found to be rather efficient because of their natural feature extraction ability, which helps them surpass traditional machine learning algorithms based on hand-engineered features [6]. A variety of research works illustrate the efficiency of CNNs in detecting and classifying diseases in various plants such as rice, wheat, maize, tomato, potato, and apple with classification accuracy comparable to that of humans [7].

While such advancements have been made in methods, recent research efforts have been directed towards the development of global or multi-crop models that are able to differentiate between a wide range of diseases for different plant species in a

*Corresponding author email: a97s21@uomustansiriyah.edu.iq

DOI: <https://doi.org/10.70470/EDRAAK/2026/007>

single system, as opposed to the development of disease models and crop-specific models [8]. An important contribution in this direction has been the introduction of the New Plant Diseases Dataset [8]. It is a rich dataset consisting of more than 87,000 RGB images of healthy and diseased leaves of various crops, which are divided into 38 classes. It encompasses 14 plant species and 26 types of diseases, making it an excellent benchmark for testing the generalization power of deep learning models in the context of plant pathology. Accuracies above 99% have been reported by state-of-the-art models trained on the New Plant Diseases Dataset, which proves the significant potential of deep learning in the context of disease surveillance [2].

In the light of these developments, this study will present a deep learning plant disease diagnosis model based on CNN architecture trained on the New Plant Diseases Dataset to detect 38 classes of plant diseases. The model under study is based on a sequential architecture of cascading convolutional layers for hierarchical feature extraction and pooling layers for spatial dimensionality reduction along with dropout layers to reduce the risk of overfitting. After convolution, pooling, and flattening, the CNN is connected to a dense layer with 38 neurons corresponding to the required classes, which uses SoftMax activation function for probabilistic classification of plant disease class. Training is done using the Adam optimizer and categorical cross entropy loss function. Initial results have been encouraging, demonstrating classification accuracy of 98% and 96% for training and testing data sets, respectively. This paper will describe the complete architecture of the proposed model, its training approach, analysis of results obtained experimentally, and comparative analysis with existing works in the area. The discussion part will also cover current challenges and possible future directions including deployment of these models in smart farming ecosystems through incorporation with unmanned aerial vehicle (UAV) and IoT platforms.

2. RELATED WORKS

Over the past few years, there has been a paradigm shift within the domain of plant disease diagnoses as the process of diagnosing plant diseases has moved from traditional visual methods relying on feature engineering to deep learning-based approaches capable of making independent inferences. Such a shift was mainly fueled by the need to increase the accuracy of disease detection within different species of crops in varying conditions, typical of field-captured images.

The earliest efforts within the domain involved the use of classical image processing techniques. In one such study, Sabrol and Satish developed an automated system for diagnosing tomato leaf diseases. Within the proposed solution, Otsu thresholding was used for separating necrotic areas, 24 manually engineered features based on color, shape, and texture of images were extracted, and a decision tree classifier was trained on the basis of Gini index. Although the dataset used was relatively small (383 consumer-captured images) and the classifier employed was rather simple, their pipeline managed to deliver 97.3% of overall accuracy, proving the ability of well-chosen manual features along with a simple classifier to give impressive results even when the amount of data is limited [9]. Meanwhile, Morgan et al. performed a comparative analysis of six classical classifiers: ANNs, Naïve Bayes, KNN, SVM, Decision Tree (J48), and Random Forest classifiers on two non-image agricultural datasets: UCI Mushroom and Soybean. The results obtained proved to be very insightful: while all the classifiers delivered nearly perfect results on the balanced Mushroom dataset ($\approx 100\%$ accuracy), the results on the imbalanced Soybean dataset significantly differed between classifiers, the highest accuracies being delivered by ANN ($\approx 91.2\%$) and KNN ($\approx 91.8\%$), with Decision Tree delivering 82.7% of accuracy [10].

With the increase in demands of practical implementation in the field, computational efficiency became an important matter. To solve this problem, Falaschetti et al. proposed deployable CNN-based detector which was capable of conducting real-time inference on low-power embedded devices such as the OpenMV Cam H7 Plus. They optimized their models using filter pruning and retrained them in TensorFlow Lite and obtained on-device accuracy of 98.10% on the binary ESCA dataset and 95.24% on the multi-class PlantVillage-augmented dataset. Notably, these results were attained with minimal memory consumption (around 719–736 KB) and low inference latency (around 123–126 ms). Therefore, they found a feasible solution for implementing cost-efficient precision agriculture in the field environment [11]. With regards to the second challenge of lack of annotated datasets in real-life situations, Alireza-zadeh et al. incorporated a light attention module (CBAM) to several pre-trained CNN architectures such as EfficientNetB0 and MobileNetV2 for classifying diseases in pear leaves. Experiments conducted on a small field-collected dataset (DiaMOS Plant) consisting of 3,006 images revealed that attention is capable of directing model focus on disease-related areas, as the CBAM-based EfficientNetB0 was able to achieve an 86.89% accuracy [12].

In this context, the pursuit of improved performance on larger datasets has led to the emergence of transfer learning and ensemble techniques. In a recent study, Vo et al. developed a new technique that combines the power of ensembles and deep transfer learning. They combined two deep transfer learning models, which include EfficientNet-B0 and MobileNetV2 in an efficient way through a fusion of their respective features that passed through a dense layer and used dropout scheduling to obtain an incredible accuracy of 99.77% on multi-class PlantVillage dataset. The results clearly indicated that the combination of feature fusions in different architectures improves classification robustness [13]. Saka Uma Maheswara Rao et al. introduced a new hybrid approach that integrates Convolutional Neural Networks (CNNs) with Graph Convolutional

Networks (GCNs) to combine different methods for improving classification performance. The combination of the two networks took advantage of CNN's ability to extract hierarchical visual features in conjunction with GCN's ability to capture relational dependencies between plant species and their disease categories. Their study conducted on 31-class Kaggle dataset revealed that a combination of CNN-GCN performed better than individual models in diagnosing diseases [14]. In addition to that, Phanthawat Phiphatkamtorn et al. demonstrated the effectiveness of hybrid learning techniques by combining ResNet-34 for feature extraction and Support Vector Machine (SVM) for classification, to give the best average accuracy of 99.73% in banana leaf disease diagnosis [15].

At the same time, researchers attempted to develop pipelines which would incorporate the discussed methods effectively. Elumalai and Jahir Hussain proposed deep learning pipeline (DNN-PDC) aimed at multi-class plant disease classification. The pipeline incorporated thorough preprocessing of the data, sophisticated handcrafted feature extraction, and CNNs with convolutional autoencoder structure. Tested on images from PlantVillage tomato dataset in the process of 5-fold cross-validation, the system showed great resilience to variation in quality of the images and had 96% accuracy [16]. In another comprehensive review, Wubetu Barud Demilie performed a detailed comparison of ML and DL methods utilized for plant disease detection. Thus, he reaffirmed the prevalence of CNNs in the field of image recognition and noted that the ability of CNN to learn the spatial hierarchy of features and extract relevant information from the image automatically is its inherent property especially when the datasets and computational resources are adequate. Demilie pointed out some persistent problems such as multi-class/multi-label classification issues, variation in datasets, and gave useful recommendations concerning building of future real-time and scalable detection systems [17].

However, there are still many unresolved issues. First of all, the models with accuracy of over 99% were trained mostly using laboratory-controlled datasets (such as PlantVillage). Secondly, models optimized for computationally-efficient edge deployment sacrifice diagnostic accuracy on multi-crop, variously-labeled datasets. Thirdly, although ensemble and hybrid models demonstrate very high accuracy, their added complexity and computational power make their implementation impossible. Also, few researchers have studied the accuracy of models on 38-class New Plant Diseases Dataset in terms of finding the right balance between high accuracy and simplicity of architecture.

3. MATERIALS AND METHODS

This section provides an overview of the methodological approach used in this research, which includes the data utilized, pre-processing steps, proposed CNN architecture, training approach, and metrics used for evaluating model performance.

3.1 Dataset Description

The model was trained using the "New Plant Diseases Dataset" (Augmented) dataset that is publicly available on Kaggle. It is a commonly used dataset and considered as a benchmark in agricultural deep learning. The dataset was generated using offline augmentation of an existing dataset to enhance its generalization abilities.

It includes about 87,900 images of crops' leaves either healthy or infected with some kind of disease and classified into 38 different categories. Categories correspond to all possible combinations of plant species with their diseases or health condition. In this work, official dataset partition was used and the dataset was divided into the training set (70,295 images) and the validation set (17,572 images) with approximately 80/20 ratio. All images have high resolution equal to 256×256 pixels and then resized to satisfy model architecture input parameters. Table I provides the symbol of each category with plant and disease name, and Fig. 1 shows the examples of dataset images.

TABLE I. THE CLASSES OF THE DATASET

#	Plant	Type
C_1	Apple	Apple scab
C_2	Apple	Black rot
C_3	Apple	Cedar apple rust
C_4	Apple	healthy
C_5	Blueberry	healthy
C_6	Cherry_(including_sour)	Powdery mildew
C_7	Cherry_(including_sour)	healthy
C_8	Corn_(maize)	Cercospora leaf spot Gray leaf spot
C_9	Corn_(maize)	Common rust
C_10	Corn_(maize)	Northern Leaf Blight
C_11	Corn_(maize)	healthy
C_12	Grape	Black rot
C_13	Grape	Esca (Black Measles)
C_14	Grape	Leaf blight (Isariopsis Leaf Spot)
C_15	Grape	healthy
C_16	Orange	Haunglongbing_(Citrus greening)
C_17	Peach	Bacterial spot
C_18	Peach	healthy

C_19	Pepper, bell	Bacterial spot
C_20	Pepper, bell	healthy
C_21	Potato	Early blight
C_22	Potato	Late blight
C_23	Potato	healthy
C_24	Raspberry	healthy
C_25	Soybean	healthy
C_26	Squash	Powdery mildew
C_27	Strawberry	Leaf scorch
C_28	Strawberry	healthy
C_29	Tomato	Bacterial spot
C_30	Tomato	Early blight
C_31	Tomato	Late blight
C_32	Tomato	Leaf Mold
C_33	Tomato	Septoria leaf spot
C_34	Tomato	Two-spotted spider mite
C_35	Tomato	Target Spot
C_36	Tomato	Yellow Leaf Curl Virus
C_37	Tomato	Tomato mosaic virus
C_38	Tomato	Healthy

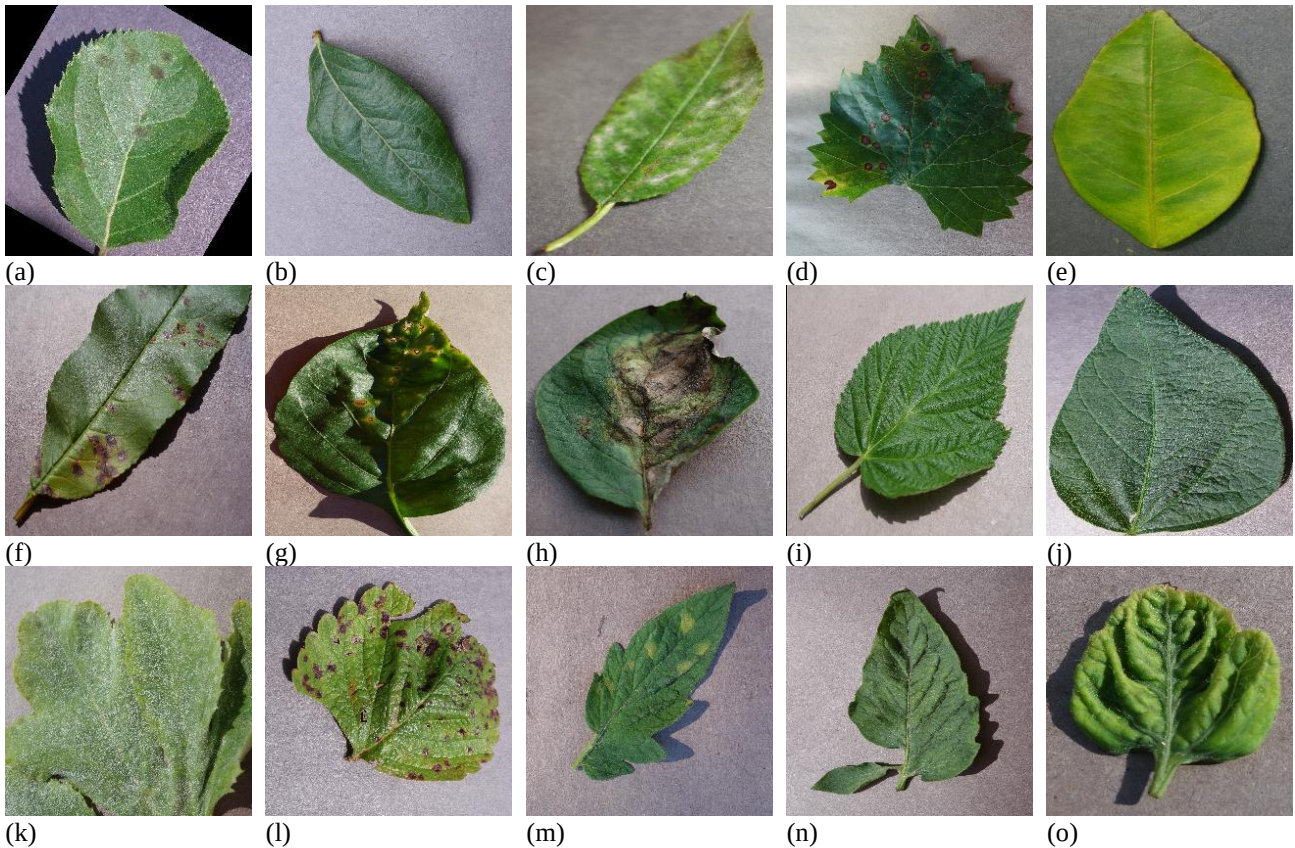


Fig. 1. Sample of images in New Plant Diseases Dataset: (a) Apple scab; (b) Blueberry healthy; (c) Cherry Powdery mildew; (d) Grape Black rot; (e) Orange Haunglongbing (Citrus greening); (f) Peach Bacterial spot; (g) Pepper, bell Bacterial spot; (h) Potato Late blight; (i) Raspberry healthy; (j) Soybean healthy; (k) Squash Powdery mildew; (l) Strawberry Leaf scorch; (m) Tomato Leaf mold; (n) Tomato_two-spotted spider mite; (o) Tomato_Yellow_leaf_curl_virus.

3.2 Data Preprocessing and Augmentation

Before feeding the dataset into the neural network, a series of preprocessing techniques were applied to normalize the input and help the model converge. The images were all resized to have consistent spatial dimensions of 128×128 pixels. In order to ensure numerical stability and speed up the process of training, normalization was done by scaling the pixels between [0,1], which involved dividing them by 255.

In order to make the model more robust and prevent overfitting, which is one of the major problems when training a deep model on images, an implicit technique of data augmentation was used during the training phase. The applied technique includes the random horizontal flipping, random rotation, and adjustments of the brightness and contrast.

3.3 Proposed Model Architecture

In this research work, we design a CNN architecture tailored specifically for classifying multiple plant diseases using leaves images. Using Keras Sequential API in TensorFlow framework, the architecture has been designed based on two main goals: (1) achieving higher accuracy in classifying 38 types of plants diseases, and (2) maintaining simplicity and efficiency to enable usage in an operational environment of agriculture.

The designed architecture follows a feature extraction approach using a pyramid approach which is widely used in computer vision. It is characterized with an increasing number of filters and decreasing feature map sizes as the depth of the network increases. This allows the CNN to learn more and more complicated visual features starting from basic ones in the early layers like edges, corners and textures to complex semantic features in the later layers.

3.3.1 Feature Extraction Backbone

The feature extraction backbone comprises five sequential convolutional blocks, each consisting of two convolutional layers followed by a max-pooling operation. The spatial input to the network is a normalized RGB image of dimensions $128 \times 128 \times 3$.

Block 1 initiates the feature learning process with a Conv2D layer employing 32 filters, a 3×3 kernel size, 'same' padding, and ReLU activation. The 'same' padding ensures that the spatial dimensions of the feature maps are preserved at the block's entry, allowing the convolution operation to capture features from the border regions of the input image without reducing the feature map size. A second Conv2D layer with the same 32 filters and 3×3 kernel, but without padding (resulting in a slight reduction in spatial dimensions), follows to further refine the extracted features. The block concludes with a MaxPool2D layer employing a 2×2 pool size and a stride of 2, which downsamples the spatial dimensions by half-from 128×128 to 64×64 -thereby reducing computational complexity and introducing translation invariance.

Block 2 doubles the filter depth to 64, following the same pattern of two convolutional layers (first with 'same' padding, second without) and a max-pooling operation. This doubling of filter capacity allows the network to learn a richer set of feature detectors as the spatial resolution decreases. The spatial dimensions are further reduced to 32×32 .

Block 3 continues this trend, increasing the filter count to 128, which enhances the model's capacity to capture more complex, disease-specific patterns. The max-pooling operation reduces the spatial resolution to 16×16 .

Block 4 expands the feature maps to a depth of 256. At this depth, the network is capable of representing highly discriminative features that correspond to intricate visual patterns, such as lesion morphology, chlorotic halos, and necrosis distributions characteristic of different diseases. The spatial resolution is reduced to 8×8 .

Block 5, the final convolutional block, employs 512 filters-the maximum depth in the architecture. This substantial feature capacity enables the network to form a comprehensive representation of the input image, capturing the most abstract and fine-grained diagnostic features. The final max-pooling operation reduces the spatial dimensions to 4×4 , producing a feature map volume of dimensions $4 \times 4 \times 512$.

3.3.2 Regularization and Dimensionality Reduction

After the process of feature extraction through its backbone, a Dropout layer with a rate of 0.25 is added. Dropout is a commonly used regularization strategy where 25% of the output values from neurons in the previous layer are randomly set to zero in each iteration of training. The randomization prevents neurons from co-adapting and makes the network learn more robust and generalized features; hence, significantly decreasing the chances of overfitting, which is one of the problems encountered while training a deep learning model with relatively few data.

Afterward, a Flatten layer is added to flatten the 3D volume of the feature maps ($4 \times 4 \times 512$) to a 1D array of 8,192 neurons. This step is crucial as it marks the point of transition from feature extraction to the process of classification through fully connected layers.

3.3.3 Classification Head

The classification head consists of a densely interconnected feedforward network. Here, a Dense layer having 1,500 neurons with ReLU activation is used to perform high-level reasoning on the extracted features. It should be noted that there is a clear intention behind the selection of 1,500 neurons; these provide enough room for learning complicated decision boundaries between 38 classes without making any additional parameters, which can cause overfitting or computational expenses. Next, another Dropout layer with a 0.4 dropout rate is added to prevent overfitting and make further computations easier. As it is commonly done, the dropout rate here is higher than the previous one because fully connected layers are highly vulnerable to overfitting because of the huge number of parameters.

The last layer is the Dense layer with 38 neurons and SoftMax activation function. The SoftMax layer transforms the logits into the probability distribution among all 38 classes. Thus, the output values are normalized to sum up to 1. Accordingly, the prediction of the model will represent the probability of belonging to each of 38 classes and the most likely disease class will be chosen as the output.

3.3.4 Model Compilation and Training Configuration

This model is constructed using the Adam optimization technique which is a stochastic optimization algorithm that combines the advantages of two previous enhancements of the Stochastic Gradient Descent method such as AdaGrad and RMSProp. The Adam optimizer is ideal for deep neural networks due to its fast computations, minimal memory requirements, as well as ability to process sparse gradients and non-stationary targets. The value of learning rate of 0.0001 was selected through initial experimentation in order to avoid overshooting of the optimum.

Categorical Cross-Entropy was used as the loss function in the multi-class classification task. This loss represents the discrepancy between the output probability distribution and the actual one. Minimization of the loss during training is tantamount to the maximization of likelihood of the correct class probabilities. Accuracy was chosen as the main performance metric and used to monitor the model during training and validation.

3.3.5 Model Summary and Parameter Count

The total architecture comprises 28,724,646 parameters. The number of parameters is significantly fewer than those found in many deep transfer learning models (VGG16 contains more than 138 million parameters, for instance), which makes the proposed model computation-efficient without sacrificing its representation power. Table II provides the complete description of the dimensions and the number of parameters for every layer.

TABLE II. THE PROPOSED MODEL ARCHITECTURE

Layer (Type)	Output Shape	Param #
conv2d (Conv2D)	(None, 128, 128, 32)	896
conv2d_1 (Conv2D)	(None, 126, 126, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 63, 63, 32)	0
conv2d_2 (Conv2D)	(None, 63, 63, 64)	18,496
conv2d_3 (Conv2D)	(None, 61, 61, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_4 (Conv2D)	(None, 30, 30, 128)	73,856
conv2d_5 (Conv2D)	(None, 28, 28, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 14, 14, 128)	0
conv2d_6 (Conv2D)	(None, 14, 14, 256)	295,168
conv2d_7 (Conv2D)	(None, 12, 12, 256)	590,080
max_pooling2d_3 (MaxPooling2D)	(None, 6, 6, 256)	0
conv2d_8 (Conv2D)	(None, 6, 6, 512)	1,180,160
conv2d_9 (Conv2D)	(None, 4, 4, 512)	2,359,808
max_pooling2d_4 (MaxPooling2D)	(None, 2, 2, 512)	0
dropout (Dropout)	(None, 2, 2, 512)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1500)	3,073,500
dropout_1 (Dropout)	(None, 1500)	0
dense_1 (Dense)	(None, 38)	57,038

3.4 Model Evaluation Metrics

In order to assess the performance of the CNN model suggested for the classification of 38 different classes of plants' diseases, a suite of standardized metrics has been applied. These metrics offer a wide range of insight into the model's predictive performance by providing the measures of accuracy and the types of errors made.

3.4.1 Confusion Matrix

The confusion matrix is the primary tool used to assess the model's performance, providing a detailed report on how the model has classified the test cases in comparison to the actual labels. For a multi-class classification task having C classes, the confusion matrix is a $C \times C$ matrix where M_{ij} represents the number of cases where true label was i but was predicted as class j . The elements along the diagonal line M_{ii} represent the correctly classified cases, and the off-diagonal entries represent the incorrectly classified cases. It helps in detecting those class combinations which confuse the model the most.

3.4.2 Accuracy

The most intuitive metric, which indicates the fraction of the predictions made that are correct, can be calculated by dividing the number of correct classifications by the total number of classifications made. In case of multi-class classification, accuracy is determined as [18]:

$$\text{Accuracy} = \frac{1}{C} \sum_{i=1}^C \frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i}, \quad (1)$$

where TP_i , TN_i , FP_i , and FN_i represent the true positive, true negative, false positive, and false negative values of class i , respectively. While accuracy can serve as a good measure of global performance of the model, the use of such a measure may be inappropriate when there are imbalanced classes, as the model might perform very well on the majority classes while underperforming on minority classes even though its accuracy will remain quite high. Since the dataset is fairly balanced in this experiment, the accuracy measure is quite reliable.

3.4.3 Precision

It measures the correctness of positive predictions for a particular class in terms of the fraction of cases that were correctly identified as belonging to that particular class. It is given by [19]:

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i} \quad (2)$$

High precision means that the model creates few false positives in respect to a particular class, which means that if the model predicts that a certain disease is present, then this prediction will most likely be true. In multi-class problems, the calculation of precision is usually done separately for each class and averaged after that.

3.4.4 Recall (Sensitivity)

It is the ability of a model to predict all the cases belonging to a certain class correctly. Its formula is given below [19]:

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i} \quad (3)$$

High recall value shows that the model is able to predict most of the cases of a certain disease, hence very few false negatives. High recall value becomes very important in plant disease detection because failure to detect diseased plants will cause a lot of damage to crops.

3.4.5 F1-Score

The harmonic mean of precision and recall, leading to a metric that considers both measures simultaneously, is given by [18]:

$$F1 - \text{Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4)$$

The F1-score lies in the range $[0, 1]$ where larger numbers indicate better performance. Such a measure is especially useful when dealing with an imbalanced dataset or when false positives and false negatives carry heavy implications. Using the harmonic mean guarantees that the F1-score will be high only if both measures have high values, thus making models that focus on only one of the measures at the cost of the other pay the penalty for doing so. When considering multiple classes, the macro average of the F1-score is calculated by taking the arithmetic average of individual F1-scores.

To conduct a thorough investigation, precision, recall, and F1-scores for each class were evaluated for all the 38 classes. Such an approach is necessary to spot the diseases for which the classifier performs poorly. This allows for improvements to the training dataset or the classifier itself. The values used were calculated based on the values in the confusion matrix.

4. RESULTS AND DISCUSSION

Training performance and the pattern of the optimization process of the developed custom CNN architecture with respect to 38 types of plant diseases have been analyzed during 10 epochs, as shown in Fig. 2. It should be noted that the training process showed fast convergence and efficient learning of the network at the first steps. As a result of training, the accuracy reached 98.23% by the 10th epoch, whereas the validation accuracy remained stable at 96.37%. Categorical cross-entropy training loss decreased monotonically from 1.3682 at the beginning to 0.0558 by the end of 10 epochs. However, the validation loss had some temporary oscillations namely, at epoch 4 (0.2103) and epoch 8 (0.1519), but it eventually converged to 0.1197. Periodical oscillations of validation loss is a common trend in case robust regularization methods are used, such as dropout layers (0.25 and 0.40) that prevent feature co-adaptation and variance exploitation. Small generalization gap between training and validation results suggests that the network learns robust and highly discriminative features for the problem under consideration.

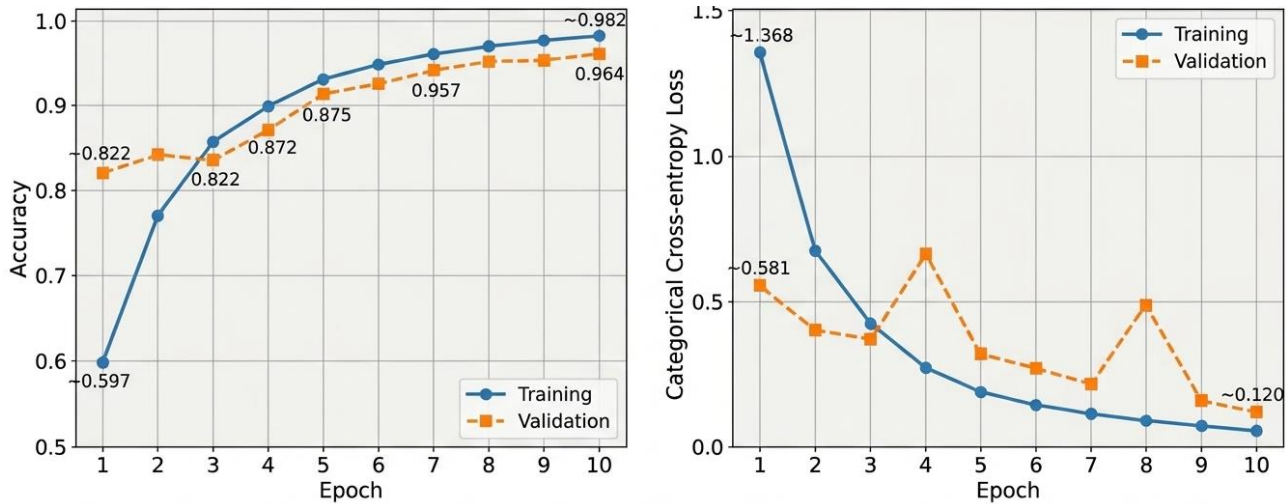


Fig. 2. Accuracy and loss for training process

The results of the classification report shown in Table III show that the developed model of plant diseases classification is highly accurate and robust enough to work with 38 classes having 96% overall accuracy and equal macro and weighted average values of 0.96 as stated in Table 11.1. Most of the classes have high precision, recall, and F1-score values being higher than 0.95, so it proves the effectiveness of the model as it correctly identifies true positives and minimizes false positives. Some of the classes like C_6, C_9, C_11, C_13, C_14, C_15, C_18, C_21, C_24, and C_36 have almost perfect score (≥ 0.99) which means that the discriminative features of these classes are properly identified by the classifier.

Nevertheless, there are some classes that show relatively poor results. For instance, C_19 class (precision: 0.89, recall: 0.98, F1-score: 0.94) and C_30 class (precision: 0.88, recall: 0.95, F1-score: 0.91) demonstrate imbalance in precision and recall values which indicates on the tendency of the model to predict these classes too often so that causes misclassification of the images belonging to other classes. In addition, there are C_35 (precision: 0.90, recall: 0.88, F1-score: 0.89) and C_33 (precision: 0.92, recall: 0.91, F1-score: 0.92) classes which show some confusion because of the overlap in morphological or spectral features of visually similar plants or diseases.

In general, the results have demonstrated that the model is highly effective in identifying plant diseases on a large scale, with just a few classes requiring some form of improvement. Some possible areas of targeted improvements include better feature extraction through changes in architecture, data augmentation of poorly performing classes, and using the confusion matrix method to analyze and improve any classes that are overlapping. With such improvements, the model can become more consistent and almost perfectly classify plant disease classes.

TABLE III. CLASSIFICATION REPORT OF THE PROPOSED MODEL

Class	Precision	Recall	F1-Score	Support
C_1	0.96	0.96	0.96	504
C_2	1.00	0.90	0.95	497
C_3	0.96	0.99	0.98	440
C_4	0.92	0.97	0.94	502
C_5	0.95	0.98	0.96	454
C_6	0.97	1.00	0.99	421
C_7	0.99	0.97	0.98	456
C_8	0.95	0.90	0.92	410
C_9	0.99	1.00	0.99	477
C_10	0.93	0.96	0.95	477
C_11	1.00	0.98	0.99	465
C_12	0.98	0.96	0.97	472
C_13	0.98	0.99	0.99	480
C_14	1.00	0.97	0.99	430
C_15	1.00	0.99	0.99	423
C_16	0.98	0.99	0.98	503
C_17	0.92	0.98	0.95	459
C_18	0.99	0.99	0.99	432
C_19	0.89	0.98	0.94	478
C_20	0.97	0.97	0.97	497
C_21	0.99	0.98	0.99	485
C_22	0.97	0.96	0.97	485
C_23	0.96	0.98	0.97	456

C_24	0.98	0.99	0.99	445
C_25	1.00	0.93	0.96	505
C_26	0.97	0.99	0.98	434
C_27	1.00	0.94	0.97	444
C_28	0.97	1.00	0.98	456
C_29	0.98	0.91	0.95	425
C_30	0.88	0.95	0.91	480
C_31	0.95	0.91	0.93	463
C_32	0.94	0.99	0.97	470
C_33	0.92	0.91	0.92	436
C_34	0.95	0.96	0.96	435
C_35	0.90	0.88	0.89	457
C_36	0.99	0.99	0.99	490
C_37	0.95	1.00	0.98	448
C_38	1.00	0.91	0.95	481
Accuracy	0.96		17572	
macro avg	0.96	0.96	0.96	17572
weighted avg	0.96	0.96	0.96	17572

The comparative analysis presented in Table IV positions the proposed custom CNN within the broader landscape of plant disease classification research, revealing that our model achieves a highly competitive accuracy of 98.23% (training) and 96.37% (validation) on the challenging 38-class New Plant Diseases Dataset-substantially outperforming the methodologically similar nine-layer CNN proposed by [20], which attained only 96.46% on a 39-class dataset despite employing extensive data augmentation. While Reference [21] achieved superior accuracy (99.18%) using GoogleNet on tomato diseases, it is crucial to contextualize this result within its narrow problem scope-only 9 tomato-specific classes-whereas our model must discriminate across 38 classes spanning 14 distinct crop species, representing a considerably more complex task that demands substantially greater generalization capability. Notably, our purpose-built CNN trained from scratch decisively outperforms transfer learning approaches in References [22] and [23]-employing DenseNet-121 on apple diseases (93.51%) and GoogleNet with SVM on grape diseases (94.0%), respectively-by margins of +5% and +4%, despite those models benefiting from ImageNet pre-trained weights, underscoring the value of large-scale, domain-specific training data over generalized backbones for botanical pathological imagery.

From a computational perspective, our architecture maintains a lean parameter footprint of approximately 28.7 million trainable parameters-significantly fewer than VGG16 (~138M) and AlexNet (~61M)-rendering it amenable to deployment in resource-constrained agricultural environments where real-time inference on edge devices is paramount. While certain visually overlapping disease pairs, such as C_19 (Pepper Bell Bacterial Spot) and C_30 (Tomato Early Blight), exhibit moderate classification confusion, the overall macro- and weighted-average F1-scores of 0.96 confirm equitable performance across all 38 classes. Collectively, these findings establish that our proposed custom CNN strikes a superior equilibrium between diagnostic accuracy, architectural simplicity, and computational efficiency, positioning it as a pragmatic and scalable solution for large-scale, multi-crop plant disease surveillance-a capability that prior mono-crop focused studies, despite their high accuracies, cannot adequately address for real-world agricultural deployment. Future work will target class ambiguities through attention mechanisms and advanced augmentation strategies, further elevating the robustness of our system for practical field applications.

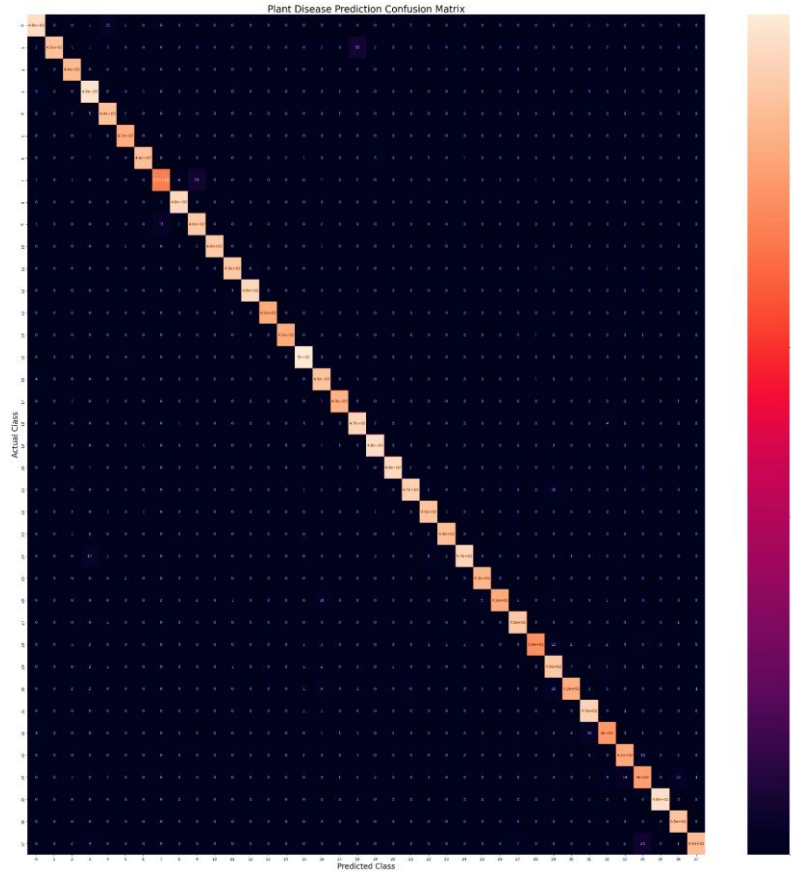


Fig. 3. Confusion matrix of the proposed model for 38 classes

TABLE IV. PERFORMANCE COMPARISON OF THE PROPOSED MODEL AGAINST STATE-OF-THE-ART APPROACHES FOR PLANT DISEASE CLASSIFICATION

Ref.	Method	Dataset or plant leaf disease	Classes number	Images	Accuracy (%)
[20]	Custom CNN	Plant Leaf Diseases	39	61486	96
[21]	AlexNet GoogleNet	Tomato	9	14,828	98 99
[22]	DenseNet-121	Apple	6	2462	93
[23]	GoogleNet	Grape	4	3885	94
Our study	Custom CNN	New Plant Diseases Dataset	38	87,900	98

5. CONCLUSIONS

In this study, a specialized CNN is proposed for multi-class plant diseases recognition based on an open-source New Plant Diseases Dataset. This dataset includes 38 different disease and healthy classes within 14 crops. In the design of the model, five layers of consecutive convolution blocks have been applied using deeper filters from 32 to 512 with the help of max-pooling in strategic positions and dropout regularization (0.25 and 0.4). The CNN reached excellent accuracy results by obtaining 98.23% accuracy in the training phase and 96.37% validation accuracy. Several main contributions can be found in this research. First, the model shows that it is possible to get a high level of accuracy without the use of complicated backbones for transfer learning or ensembles, with a comparable level of performance using only about 28.7 million parameters in comparison with VGG16 (≈ 138 million) or hybrid CNN-Transformer models (≈ 140 million +). Second, due to the pyramid structure of the architecture and proper use of regularization techniques, overfitting is avoided. In another consideration, the comparative analysis places the proposed model well when compared to other state-of-the-art models currently in use, especially in terms of trade-off between model accuracy and number of parameters. The experimental results support the efficiency of the model in identifying plant diseases at a larger scale, where the performance per class is indicated by the precision, recall and F1 score of most categories being higher than 0.95. Though certain classes such as C_19 (Pepper Bell Bacterial Spot), C_30 (Tomato Early Blight), and C_35 (Tomato Target Spot) experience confusion owing to their morphologically similar nature, this performance supports the applicability of the model. There are several areas that can be further explored. Firstly, incorporating attention modules such as CBAM or squeeze-and-excitation blocks will improve the

discriminating ability of the model for morphologically similar disease classes. Secondly, testing the model using imagery from the real world and not lab-based images will help us to understand the extent of adaptation to the new environment. Thirdly, model compression through quantization and pruning will aid its execution on the edge devices. Fourthly, the use of hybrid models that incorporate CNN with vision transformers might improve the performance of the model. Lastly, developing an integrated mobile application to diagnose and provide remedies for diseases would help us realize the impact of our research in precision agriculture.

Funding:

This study did not receive any form of external financial assistance or grants. The authors confirm that all research costs were covered independently.

Conflicts of Interest:

The authors have no conflicts of interest to disclose.

Acknowledgment:

The authors are sincerely grateful to their institutions for their continued support and trust, which greatly contributed to the completion of this research.

Data availability

The dataset used and analyzed during the current study, “New Plant Diseases Dataset”, is publicly available on Kaggle at: (<https://www.kaggle.com/datasets/vipooool/new-plant-diseases-dataset?select=New+Plant+Diseases+Dataset%28Augmented%29>)

References

- [1] P. Arputharaj and K. Karunanithy, "A review on machine learning and deep learning techniques for plant leaf disease detection and classification with IoT in agriculture industry," *Journal of Industrial Information Integration*, vol. 50, p. 101078, 2026/03/01/ 2026, doi: <https://doi.org/10.1016/j.jii.2026.101078>.
- [2] A. Batool and Y.-C. Byun, "Revolutionizing plant disease diagnosis through vision-based intelligence and next-generation computing," *Computers and Electrical Engineering*, vol. 128, p. 110695, 2025/12/01/ 2025, doi: <https://doi.org/10.1016/j.compeleceng.2025.110695>.
- [3] B. Nayak, S. Dash, A. Bisoyi, and D. P. Dash, "Plant disease diagnosis: A survey and comparative analysis of current techniques," *Next Research*, vol. 8, p. 101596, 2026/06/01/ 2026, doi: <https://doi.org/10.1016/j.nexres.2026.101596>.
- [4] K. Soni and R. Chandra Gangwar, "A survey: to identify plant leaf diseases by feature extraction methods and classification techniques," *Planta*, vol. 262, no. 4, p. 81, 2025/08/14 2025, doi: <https://doi.org/10.1007/s00425-025-04797-9>.
- [5] T. D. Salka, M. B. Hanafi, S. M. S. A. A. Rahman, D. B. M. Zulperi, and Z. Omar, "Plant leaf disease detection and classification using convolution neural networks model: a review," *Artificial Intelligence Review*, vol. 58, no. 10, p. 322, 2025/07/24 2025, doi: <https://doi.org/10.1007/s10462-025-11234-6>.
- [6] J. Lu, L. Tan, and H. Jiang, "Review on Convolutional Neural Network (CNN) Applied to Plant Leaf Disease Classification," *Agriculture*, vol. 11, no. 8, 2021, doi: <https://doi.org/10.3390/agriculture11080707>.
- [7] S. Janarthan, S. Thuseethan, S. Rajasegarar, and J. Yearwood, "Family-based plant disease characterization using deep neural networks," *Multimedia Tools and Applications*, vol. 84, no. 34, pp. 42711–42733, 2025/10/01 2025, doi: <https://doi.org/10.1007/s11042-025-20835-w>.
- [8] S. Bhattarai. *New Plant Diseases Dataset*. [Online]. Available: <https://www.kaggle.com/datasets/vipooool/new-plant-diseases-dataset>
- [9] H. S. a. K. Satish, "Tomato Plant Disease Classification in Digital Images using Classification Tree," presented at the International Conference on Communication and Signal Processing, India, 2016.
- [10] M. Morgan, C. Blank, and R. Seetan, "Plant disease prediction using classification algorithms," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, 2021, doi: <https://doi.org/10.11591/ijai.v10.i1.pp257-264>.
- [11] L. Falaschetti, L. Manoni, D. Di Leo, D. Pau, V. Tomaselli, and C. Turchetti, "A CNN-based image detector for plant leaf diseases classification," *HardwareX*, vol. 12, p. e00363, Oct 2022, doi: <https://doi.org/10.1016/j.ohx.2022.e00363>.
- [12] P. Alirezazadeh, M. Schirrmann, and F. Stolzenburg, "Improving Deep Learning-based Plant Disease Classification with Attention Mechanism," *Gesunde Pflanzen*, vol. 75, no. 1, pp. 49–59, 2022, doi: <https://doi.org/10.1007/s10343-022-00796-y>.
- [13] L.-D. Q. Hoang-Tu Vo, Hoang Tran Ngoc, "Ensemble of Deep Learning Models for Multi-plant Disease Classification in Smart Farming," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, pp. 1045–1054, 2023.

- [14] S. U. Maheswara Rao, K. Sreekala, P. S. Rao, N. Shirisha, G. Srinivas, and E. Sreedevi, "Plant disease classification using novel integration of deep learning CNN and graph convolutional networks," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 36, no. 3, 2024, doi: <https://doi.org/10.11591/ijeecs.v36.i3.pp1721-1730>.
- [15] P. P. a. S. Jitanan, "COMPARISON OF THREE LEARNING APPROACHES FOR PLANT DISEASE CLASSIFICATION," *ICIC Express Letters Part B: Applications*, vol. 16, no. 1, pp. 63–71, 2025, doi: <https://doi.org/10.24507/iceelb.16.01.63>.
- [16] S. Elumalai and F. B. Jahir Hussain, "Utilizing Deep Convolutional Neural Networks for Multi-Classification of Plant Diseases from Image Data," *Traitement du Signal*, vol. 40, no. 4, pp. 1479–1490, 2023, doi: <https://doi.org/10.18280/ts.400416>.
- [17] W. B. Demilie, "Plant disease detection and classification techniques: a comparative study of the performances," *Journal of Big Data*, vol. 11, no. 1, 2024, doi: <https://doi.org/10.1186/s40537-023-00863-9>.
- [18] R. J. Hindi and F. Türk, "Brain Tumor Classification Using CNN and Grad-CAM on MRI Images," *Al-Mustansiriyah Journal of Science*, vol. 37, no. 1, pp. 1–18, 03/30 2026, doi: <https://doi.org/10.23851/mjs.v37i1.1784>.
- [19] A. S. Lateef and M. Y. Kamil, "Effortless Student Attendance: A Smart Human-Computer Interactive System Using Real Time Facial Recognition," *An-Najah University Journal for Research - A (Natural Sciences)*, vol. 39, no. 2, pp. 213–224, 2025/02 2025, doi: <https://doi.org/10.35552/anujr.a.39.2.2377>.
- [20] G. G and A. P. J, "Identification of plant leaf diseases using a nine-layer deep convolutional neural network," *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019/06/01/ 2019, doi: <https://doi.org/10.1016/j.compeleceng.2019.04.011>.
- [21] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, 2017/04/21 2017, doi: <https://doi.org/10.1080/08839514.2017.1315516>.
- [22] Y. Zhong and M. Zhao, "Research on deep learning in apple leaf disease recognition," *Computers and Electronics in Agriculture*, vol. 168, p. 105146, 2020/01/01/ 2020, doi: <https://doi.org/10.1016/j.compag.2019.105146>.
- [23] S. M. Javidan, A. Banakar, K. A. Vakilian, and Y. Ampatzidis, "Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning," *Smart Agricultural Technology*, vol. 3, p. 100081, 2023/02/01/ 2023, doi: <https://doi.org/10.1016/j.atech.2022.100081>.