Journal of Harbin Engineering University
ISSN: 1006-7043

Vol 44 No. 12
December 2023

Implementation of Deep Learning Approach with Attention Mechanism for Multiclassification for Plant Disease Detection

[1]Garima Joshi [2]Prashant Panse
[1]Medicaps University
[2]Medicaps University

Abstract
Introduction: The agricultural industry is greatly affected by the essential job of disease identification from leaf images of plants. The ability to recognize these diseases through a simple interface or machine learning model can empower farmers with better preparation strategies.
Objectives: The goal of this study is to build a Convolutional Neural Network (CNN) model that can identify plant diseases more reliably.The model leverages the MobileNetV2 network and incorporates an attention mechanism to improve feature extraction and prediction accuracy.
Methods: The proposed CNN model was built on the MobileNetV2 architecture, with modifications to include an attention mechanism for better feature extraction. The study experimented with increasing the number of layers in the CNN and tested various activation functions, including hard-sigmoid, to determine their impact on the model's specificity, sensitivity, F-measure, recall, and Matthews correlation coefficient.
Results: The introduction of an attention mechanism and the increase in CNN layers significantly enhanced model performance. For tomato plants, specificity improved from 0.47 to 0.99, and accuracy increased from 71% to 99%. Apple plants saw accuracy improvements from 62% to 98%. Considering specificity, sensitivity, F-measure, recall, and Matthews correlation coefficient, among the activation functions that were investigated, hard-sigmoid fared the highest.
Conclusions: The study demonstrates the effectiveness of deep learning, particularly CNNs enhanced with attention mechanisms, in accurately identifying plant diseases from leaf images. The advancements in model architecture and activation function selection significantly improve prediction accuracy, offering a powerful tool for agricultural disease management.

Keywords: plant disease.machine learning.convolutional neural network multi-classification. deep learning.

1. Introduction

Plant pathogens affecting the fields' crops. Plant pathogens can cause severe damage when they affect a substantial agricultural yield. The majority of plant diseases exhibit a high degree of specificity towards a single plant species or group[1]. The incidence and severity of plant diseases change over different seasons, dependent upon the existence of the pathogen, environmental circumstances, and the cultivated crops and variations. Certain plant cultivars are prone to disease outbreaks, whereas others exhibit greater resistance[2]. Infectious plant diseases are often caused by pathogenic organisms such as fungus, bacteria, viruses, protozoa, insects, and parasitic plants. Plant diseases are predominantly induced by pathogenic microorganisms, including parasitic plants, fungi, bacteria, viruses, and protozoa[3]. The rising incidence of plant diseases has emerged as a substantial determinant impacting both agricultural productivity and economic efficacy[4].

Flow cytometry, polymerase chain reaction, gas chromatography-mass spectrometry, enzyme linked immunosorbent assay, and immunofluorescence are some of the laboratory-based methods used to diagnose plant diseases in the past [5]. There are constraints on the use of these approaches for the early diagnosis, management, and control of plant diseases. Because of the time and effort needed, and the intricacy of the processes[6]. Consequently, new strategies were implemented to detect plant diseases, incorporating methods based on machine learning. The presence of automatic plant disease detection systems is critical in order to quickly forecast plant diseases, thereby mitigating agricultural losses[7]. There are a number of machine learning models that are commonly used to detect plant diseases. Some examples of these models include Support Vector Machine (SVM), Random Forest, and Multiple Twin SVM (MTSVM). Deep learning models such as Convolutional Neural Networks (DCNN), ResNet (RNet), GoogLeNet, DenseNet (DNet), LeafNet (LN), and LeNet are also commonly used. It is clear that CNN, VGG, and ResNet are the most effective deep learning models for disease detection in leaves[8]. There are a number of ways to categorise plant diseases, such as binary, multiclass, and multi label classification. As a supervised machine learning problem, binary classification entails sorting data into two different groups. Data may be grouped into three or more unique categories using multiclass classification, a supervised machine learning approach. Each input sample may be given zero or more labels in multilabel classification, a supervised machine learning method [9].In all these classificationsmostly, multiclass classification used to identify the plant diseases. Leaf mould, mosaic virus, bacterial spot, and yellow leaf curl virus are just a few examples of the various plant diseases that may be classified using a multiclass classifier. The model would acquire the ability to recognise specific characteristics that are linked to each disease type. After the model has been trained, it may be used to accurately categorise new photos into the appropriate disease group[10]. Therefore, in the current study multi class classification method was applied to identify plant diseases.

2. Literature Review

In their paper, Wang et al. (2023) provide a simplified model of a deep convolutional neural network (CNN) that uses deep features in conjunction with traditional handmade local binary pattern (LBP) features to accurately classify and detect early stages of plant leaf illnesses. The suggested approach shows promise as a suitable option for plant disease management, with improved validation and test accuracies on three publicly available datasets (Apple Leaf, Tomato Leaf, and Grape Leaf)[11].

Li et al. (2022) classified plant illnesses using images of leaves using a CNN-based multiclass plant EnsembleNet (MCPE). An EnsembleNet comprising four convolutional neural units (CNNs) and a new activation function called concatenated dynamic ReLU are used in this method to improve the precision of multiclass plant disease identification. Overall, this algorithm showed 97.5 % accuracy, it surpasses the most advanced techniques in this field[12].

When it comes to real-time identification of various plant diseases, the CDCNN model outperforms both traditional machine learning and deep learning models. When evaluated on a local cotton leaf database, the CDCNN achieved a 99 percent accuracy rate, demonstrating its superior discriminative properties for multi-class leaf disease identification across different plants [13].

In order to overcome the difficulties of long training convergence times and big model parameters, Hang et al. (2019) provide a deep learning method for disease detection and classification in plant leaves. The method incorporates a global pooling layer, an inception module, and a squeeze-and-excitation (SE) module to improve illness diagnosis. The Inception architecture enhances accuracy on the leaf disease dataset by combining feature data from the convolutional layer at different sizes. To decrease the amount of model parameters, the global average pooling layer was used instead of the completely connected layer. Compared to traditional convolutional neural networks, the proposed model achieves a test dataset accuracy of 91.7%.The approach efficiently categorises plant leaf diseases[14].

The study presents a methodology that uses deep learning to accurately discern and categorise phytopathologies using images of leaves. Regardless of variations within or across categories, the approach manages to get an astounding mean cross-validation precision of 98.68% and a mean accuracy of 97.69% on new images. The need for automated and accurate detection methods in phytopathology is being addressed by the improved technique, as traditional manual examination by experts in laboratories is seen as expensive and time-consuming. The study confirms that deep neural networks are highly effective in handling the complexities and difficulties involved in developing models trained on leaf images that show substantial differences within and across classes. The findings suggest that the suggested framework might provide a major contribution to the early detection of plant diseases, therefore reducing economic losses in the agricultural industry[15].

3. Methods

3.1 Data Collection

The dataset was sourced from the PlantVillage dataset[16] available via Kaggle[17]. This comprehensive dataset encompasses 38 distinct plant types, each representing the plant species that it belongs to with its associated diseases. In this study, primary plant types that were considered are: (a) apple, (b) grape, (c) corn, and (d) tomato. Each of these plants includes one or more disease affecting the respective plant. This provides a rich, varied dataset for our analysis. Four plants were further classified based on their disease as shown in Table 1.

Table 1. Dataset considered for each plant with their type of disease.

Plant Name	Disease Type/Healthy	Image Count
Tomato	Tomato Yellow Leaf Curl Virus	2605
Tomato	Tomato Leaf Mold	952
Tomato	Tomato Septoria leaf spot	1771
Tomato	Tomato Early blight	1000
Tomato	Tomato Tomato mosaic virus	373
Tomato	Tomato Late blight	1909
Tomato	Tomato Target Spot	1404
Tomato	Tomato Spider mites Two-spotted spider mite	1676
Tomato	Tomato healthy	1591
Tomato	Tomato Bacterial spot	2127
Apple	Apple Apple scab	630
Apple	Apple healthy	1645
Apple	Apple Cedar apple rust	275
Apple	Apple Black rot	621
Grape	Grape healthy	423
Grape	Grape Leaf blight (Isariopsis Leaf Spot)	1076
Grape	Grape Esca (Black Measles)	1383
Grape	Grape Black rot	1180
Corn	Corn (maize) Northern Leaf Blight	985
Corn	Corn (maize) healthy	1162
Corn	Corn (maize) Common rust	1192
Corn	Corn (maize) Cercospora leaf spot Gray leaf spot	513

The data set composed of varied number of images for each plant for its corresponding disease, as listed in the Table 1. Apple Cedar apple rust has the lowest number of datapoints counted as 275 images, while Tomato Yellow Leaf Curl Virus disease with 2605 images stood as maximum datapoints plant disease. Figure 1 shows the example of the example images of the Apple plant and its disease shown on its leaves.

Figure 1. Sample image of Apple plant for their diseased and healthy condition used in this study, left to right Apple Scab, Healthy, Cedar Apple Rust, Black Rot

3.2 Image Analysis

In the dataset, there were multiple images for a given plant and its corresponding disease. These images were used for the similarity index calculation analysis. The feature extraction was performed to identify patterns and similarities within each disease category. In this process, the features of image were converted into a vector. Post vectorization, mean value of vectors corresponding one class disease/healthy was taken to represent a single vector. This was represented as representative vector for the given disease type. These representative vectors were used to create a heatmap. This visual representation helped in deciphering the relationship between the different diseases and their association with the images.

The representation of the vector is shown below:

Image→Vector [[[[

```
0.    0.    0.   ... 0.    0.
  0.   ]
[          1.8324013 0.    0.   ... 0.    0.
21.95518]
[          0.    0.    0.   ... 0.    0.
  0.   ]
 ...
[          0.    9.156893  0.   ... 0.    0.
  0.   ]
[          0.    20.324734  0.   ... 0.    0.
  0.   ]
[          0.    0.    0.   ... 0.    0.
  0.   ]]

[[10.129514  0.    0.   ... 0.    0.
  0.   ]
[30.31164  0.    0.   ... 0.    0.
  29.475698]
[          0.6780374 0.    0.   ... 0.    0.
  0.]
```

3.3 Model Architecture

In this model architecture pre-trained MobileNetV2[18] model was used for the classification task. The MobileNetV2 convolutional neural network, which was trained on the extensive ImageNet dataset, serves as the fundamental model [19].The input size of the model takes (224×224) sized image with three RGB color channels. MobileNetV2 has 53 layers that is trained on ImageNet data that provides a strong feature extraction base. Later, the MobileNetV2 architecture was modified by adding the dense layer added that has 1024 neurons with ReLU activation is added for learning complex features. Post this dense layer a custom layer that includes global average pooling 2D layer was added which reduces each feature map to a single value, decreasing the model's complexity and reducing overfitting. The dimensionality reduction of the dataset after applying pooling can be explained using the equation (1).

$$x_n^m = down(x_n^{m-1}, p) \qquad (1)$$

Equation (1) may be used to estimate the output of the nth local receptive field in the mth pooling layer. Here, down denotes the down sampling function, x-n-m−1. are the feature vectors from the previous layer, and k is the pooling size. The fully connected layers were pre-connected with the flatted process where the input is converted into one dimensional output. The final dense layer has number of neurons that determined by the length of labels, with softmax activation, as shown in the equation (2).

$$softmax(z)_j = e^{(z)_j} \Big/ \sum_{k=1}^{K} e^{(z)_k} \quad (\text{for } j = 1,\ldots,K)$$

(2)

Here, (K) denotes the dimension of the (z) vector. The final layer outputs a probability distribution over the class labels, indicating the likelihood of the input belonging to each class.

3.4 Attention Mechanism

The self-attention layer is a specialised component created to enhance the model's ability to concentrate on pertinent segments of the input data (image) with greater efficiency. The system has three primary elements: (a) query (w, q), (b) key (w, k), and (c) value weights (w, v). These components are evenly initialised and may be trained via backpropagation. The system functions by calculating attention scores, which ascertain the importance of various segments of the incoming material. The scores are computed by doing matrix multiplication between the query and key, and then using a softmax function to generate a probability distribution. The ultimate result is acquired by multiplying these scores with the value. By using this method, the model is able to enhance its feature extraction capabilities by paying more "attention" to the relevant parts of the input.

3.5 Hyperparameter Tuning

The model undergoes two phases of training. Initially, the parameters of the base MobileNetV2 model are kept fixed to leverage its pre-trained ImageNet weights, focusing the training on the newly added layers. Subsequently, the fine-tuning phase involves unfreezing the last 40 layers of MobileNetV2, allowing for more extensive learning process. The learning rate is a crucial hyperparameter. Here, for fine-tuning, a significantly reduced learning rate of 0.00001 was selected. This cautious method permits incremental adaptation to the new data without erasing the pre-learned features from the ImageNet dataset.The training process spans 60 epochs, using a categorical cross-entropy loss function and measuring accuracy as the metric. Both training and validation data are fed through generators, ensuring efficient memory usage and real-time data augmentation.

3.6 Model Performance

The evaluation of the success of identifying plant diseases depends on output categories that may be classified as either binary or multiclass. The confusion matrix is a crucial instrument used in this evaluation procedure. In order to compare the actual values with the anticipated values, this matrix is essential. It offers vital insights into the model's performance.

Performance parameters used in this study are detailed below:

Sensitivity: The parameter referred to as the true positive rate is crucial in assessing the model's capacity to reliably detect diseased leaves. It quantifies the accuracy in properly identifying the real positive cases.

$$Sensitivity = \frac{\sum_{i=1}^{i=n}(TP_i)}{\sum_{i=1}^{i=n}(TP_i + FN_i)}$$

Specificity: Represented as the true negative rate, specificity measures the model's effectiveness in correctly recognizing healthy leaves.

$$Specificity = \frac{\sum_{i=1}^{i=n}(TN_i)}{\sum_{i=1}^{i=n}(TN_i + FP_i)}$$

Accuracy: The equation below demonstrates that accuracy measures the total efficacy of the model in accurately predicting outcomes, regardless of whether they are positive or negative.

$$AverageAccuracy = \frac{\sum_{i=1}^{i=n}\frac{TP_i + TN_i}{P_i + N_i}}{n}$$

Recall: The probability of detection, or recall, is the product of the total number of positive outcomes and the fraction of those outcomes that are really positive. The equation specifies the information.

$$Recall = \frac{\sum_{i=1}^{i=n}(TP_i)}{\sum_{i=1}^{i=n}(TP_i + FN_i)}$$

Matthews Correlation Coefficient (MCC): This coefficient is very valuable in tackling problems related to class inequality. It offers an equitable metric that may be applied even in cases when classes vary significantly in size.

$$MCC = (TP \times TN - FP * FN) / \sqrt{(TP + FN)(TN + FP)\,(TN + FN)}$$

F-measure: Taking the harmonic mean of recall and accuracy yields the F-measure, sometimes called the F-score. This metric is essential for assessing the trade-off between recall and accuracy, offering a more holistic perspective on the model's performance.

All of the evaluation metrics equations have the following values: TP for true positive, TN for true negative, FP for false positive, FN for false negative, P for all positives, and N for all negatives.

3.7 Activation Functions

The efficiency and efficacy of learning are heavily influenced by activation functions in neural networks. Because of the non-linear characteristics introduced by these functions, the network is able to learn intricate patterns and correlations from the input. Training speed, convergence, and the capacity to represent complicated functions are all greatly impacted by the activation function choice, which in turn affects the model's performance. In this discussion, we will delve into how various activation functions impact the performance of neural network mode. This study uses variety of scoring functions which details is shown below:

Rectified Linear Unit (ReLU): The program returns the input as is if it is positive; else, it returns zero.

$A(x) = \max(O, X)$

Sigmoid: Therefore, it is beneficial for binary classification since it transfers the values that are entered into a range that is between 0 and 1.

$A = 1/(1 + e - x)$

Soft-max: It is ideal for the output layer in classification tasks as it provides a probabilistic interpretation of different class labels. Often used in the output layer for multiclass classification.

tanh: It addresses some shortcomings of the sigmoid function by normalizing the outputs, which can lead to faster convergence. Similar to sigmoid but maps values between -1 and 1.

$\tan(x) = 2/1 + e^{-2x}$

Soft-sign: In the same vein as the tanh function, this one transforms input numbers to a range of -1 to 1. However, unlike tanh, it achieves this with a simpler, more computationally efficient formula.

Hard-sigmoidal: Unlike the classic sigmoid function, which involves exponential operations, Hard-sigmoid is computationally simpler due to its linear nature.

Soft-plus: This is a substitute for dead ReLu. Infinity to zero is the range of the output.

All the activation function mention is checked on four dataset and get their performance metrices to determine the use of the specific activation function

4. Results

4.1Image Analysis

The very initial phase of this study was to analyse the images of diseased and healthy plants. The similarity in healthy and diseased image could pose challenge to the ML model to differentiate the images. Thus, the clustering was performed and the representative image from healthy and diseased were used for calculating the similarity index. This was performed for each plant type (1) Tomato (2) Apple (3) Grape and (4) Corn. The first stage was to extract the image's features and then transform them into vector form. The characteristics from the picture were extracted using the VGG16 model.VGG16 was trained on the ImageNet dataset and thus used the weightage that are predefined.

- Load the image.
- Make feature predictions using the pre-processed image and the VGG16 model which has already been trained.
- Convert the image to an array format.
- Expand the dimensions of the image array.
- Preprocess the input image array.
- Flatten the extracted features.
- Return the flattened features.

Here, the image was loaded and converted to vector array and flattened, later the mean value of this vector was calculated. The mean values of these vectors were taken for forming the heat map. Figure 2 shows the heatmap of these vectors. As it can be detected in the heatmap for the tomato plant the maximum similarity was 60.46% which is between Tomato Bacterial spot (0) and Tomato Early blight (1). However, none of the other group shared more than 60% similarity except these two groups. This showed that images of different diseases are not similar and that made the discrimination task easier. Similarly, apple leafs also showed a strong difference among them. Grapes also did not have any major similarity found in the images of different class of the diseased images. Finally, in corn plants, it was found that two of the diseases were similar and shown 74.18% similarity with each other. Similarity above 70% reflects the challenge in discriminating these two images.

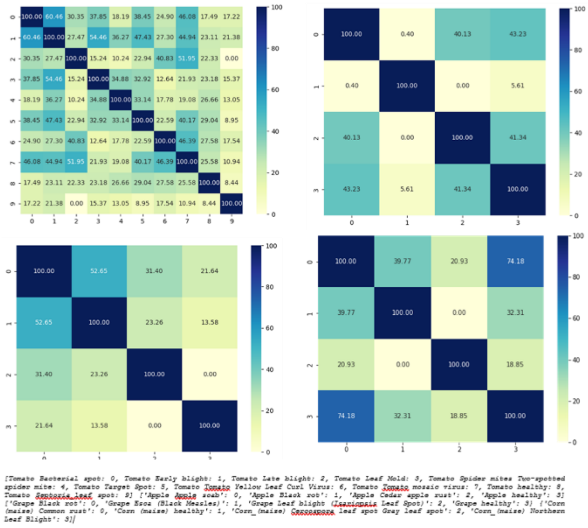

Figure 2. Heatmap of the representative image of each disease in its corresponding plant. Similarity was calculated from the image after converting them into vector.

4.2 Model Performance

The machine learning model detailed in the Method section was primarily applied on these four plant's leaf for classifying the diseases. The performance is shown in the Table 2. Sensitivity for each group for the multiclassification was '1' which means that models built in each case of the plant was capable of identifying the disease with 100% accuracy for the positive cases for each class. Specificity was recorded low compared to sensitivity of the model for each model. Tomato model showed the lowest specificity of 0.47 which means that classes in this plant was not true negatives were not predicted correctly. In tomato case, there are 10 classes including the healthy leaf, low specificity showed the issues with the model's ability to properly separate different classes, potentially due to imbalanced data, feature overlap between classes, or model complexity. Corn model showed the highest specificity with 0.98 that showed the lowest false positive rate in the prediction.

ML Models	Sensitivity	Specificity	Accuracy	Recall	F-measure	MCC
Apple Model	1	0.70	0.62	1	0.85	0.70
Tomato Model	1	0.47	0.71	1	0.70	0.47
Grape Model	1	0.81	0.80	1	0.97	0.81
Corn Model	1	0.98	0.94	1	0.99	0.98

Table 2 Model performance on the four plants considered in this study for the multiclassification

4.3 Hyperparameter Tuning

Once the base model was evaluated for their performance, as shown in the Table 2, the models were tuned for the number of layers. These layers were systemically increased and learning rate was also reduced to 0.00001. In the tuned model, the performance was improved. The result is shown in Table 3. Specificity earlier was lacking in tomato case but after hyper tuning it reached to 0.99 from 0.47. It showed how tuning affects the model's performance. In a similar vein, the model's accuracy was enhanced, and in every instance, it exceeded 95%. More dense networks have the ability to acquire hierarchical representations of features. Every layer progressively extracts more abstract characteristics from the incoming data. The use of this hierarchical representation can enhance the model's ability to differentiate between several classes, resulting in enhanced specificity.

MLModels	Sensitivity	Specificity	Accuracy	Recall	F-measure	MCC
Apple Model	1	1	0.98	1	1	1
Tomato Model	1	0.99	0.99	1	0.99	0.99
Grape Model	1	1	0.97	1	1	1
Corn Model	0.99	1	0.98	0.99	0.99	0.99

Table 3 Model performance after hyperparameter tuning on the four plants considered in this study for the multiclassification

Later, we assessed the model's accuracy for every epoch, which allowed us to establish a correlation between the epoch count, accuracy, and loss. As shown in the figure 3 that usually the accuracy increased and loss decreased with the increment in number of epochs from 1 to 60. The accuracy and loss both were calculated for train and validation set at each epoch. Both curves showed similar global trend for each plant. However, on the validation data set, the trends were marginally dissimilar. In each plant case, the training accuracy starts at around 50% and sharply increases to about 95% in the first 10 epochs, indicating rapid learning. It reached close to 100% in the final phase of the learning. On validation data, the learning was slower. This trend on validation data set is expected as the model always perform superior on the known dataset compare to the unknown dataset. In the 'loss', all the models showed rapid decrement where in settled down to zero after the initial drop down. Overall, all the graphs shown in Figure3 showed that all the models are learning efficiently and not overfitting in any case.

Application of Activation Function

Finally, the different activation functions were applied on the most optimized function. Details of the activation functions are given in the method section. Here again, sensitivity, specificity, recall, MCC, F-measure and accuracy were calculated for each activation function. Figure 4 shows the performance of these activation function. Different activation function showed variable performance on each plant case. Accuracy was detected as the most sensitive parameter that fluctuated maximally among all the evaluation metrics. In apple case, tanh, hard sigmoid and sigmoid functions showed best performance for all the metrics except accuracy. Performance of these parameters were same for this function. However, hard sigmoid performed marginally better in accuracy. Thus, it can be concluded that hard sigmoid performed slightly better than other activation functions. Hard sigmoid also performed best in the case of tomato plant as shown in Figure 3.

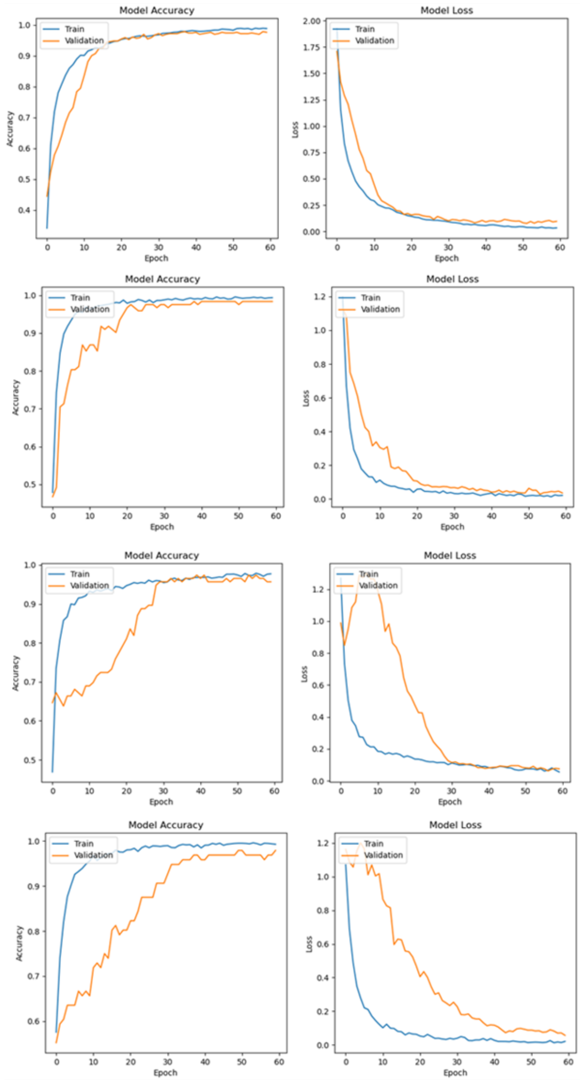

Figure 3. Training and Validation Metrics for the Machine Learning Model. The model's accuracy and loss for the training (blue) and validation (orange) datasets across 60 epochs, for (a) apples, (b) tomatoes, (c) grapes, and (d) maize, are shown on the left graph.

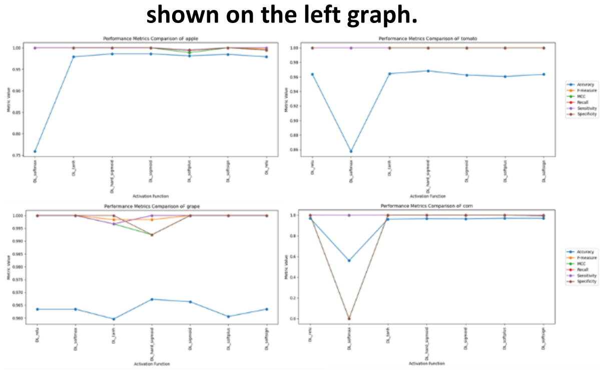

Figure 4. Validation Metrics for the Machine Learning Model using different activation functions, (a) Apple (b) Tomato (c) Grapes and (d) Corn.

Hard sigmoid also performed best for accuracy in the grape plant case, but it dipped down for other parameters. However, simple sigmoid has marginally lower accuracy but optimized score for other parameters. In the case of corn, tanh, hard sigmoid, sigmoid, softplus and softsign all showed similar performance. Overall, hard sigmoid shown the better performance in the case of all plants.

5. Discussions

Plants with different types of diseases can be detected using their images applying deep learning methods. Here, four different plants were used with their corresponding diseases. Apple, tomato, grapes and corn were considered in this study for multiclassification prediction algorithm. The dataset was sourced from PlantVillage Kaggle database. There are multiple studies reported that used this dataset for building the machine learning. Recently, to reduce the background complication from the image for the recognition a CNN model was designed based on VGG-16 network for tomato leaf disease[20]. Another article that published last year showed the importance of the machine learning on disease classification for the plant disease using the PlantVillage dataset[21]. These studies validated the application of PlantVillage dataset for building the machine learning disease prediction model. These plants are different in nature thus the type of the diseases and their manifestation on their leaves are different. This made the development of single multiclassification prediction model for all the plants, non-feasible. Thus, in this study, a multiclassification model was built for each plant. However, the framework of the models is similar for each prediction model. This study applied the attention mechanism to focus the disease patches observed on the leaf images. In previous study, attention mechanism was mentioned to identify the diseases of the plants. Support vector machine was used along with the attention mechanism to extract features for crop disease classification[22]. Channel attention mechanism was also mentioned in another study to capture the critical features for plant disease detection[23]. Attention mechanism presented in the current study also showed the improved prediction. Application of attention mechanism in other studies justified its role in the current study. Another, key argument of this study was the hyper-parameter tuning of the machine learning variables. Here, the focus was on the number of layers and learning rate. It was shown in this study that hyperparameter tuning significantly improved the prediction results. Impact of the hyperparameter tuning was shown in recent study where the case study was used for plant disease detection[24]. This study also confirmed that hyperparameter tuning is required to improve the results of prediction. Finally, activation function role was also shown to find the best activation function for the disease prediction.

6. Conclusions

This study includes four plants for building the plant disease prediction ML model using the leaf images. Different plants have different diseases and thus multiclassification model is required for the plant disease detection. Here, a CNN model proposed over the MobileNetV2 pretrained model to attain the high accuracy. This study clearly showed the conclusion of attention mechanism to improve the overall prediction accuracy. Finally, the study clearly demonstrated that increasing the number of layers that added after MobileNetV2 network increased the specificity of the classification prediction. Multiple activation functions were applied on the best model to detect the best activation function. Here, hard sigmoid performed the best outcome for maximum cases.

Refrences

[1] J. R. Ryan, “Biological Threat to Agriculture,” in *Biosecurity and Bioterrorism*, Elsevier, 2016, pp. 185–216. doi: 10.1016/B978-0-12-802029-6.00008-6.

[2] M. C. Shurtleff, A. Kelman, M. J. Pelczar, and R. M. Pelczar, “Plant disease | Importance, Types, Transmission, & Control | Britannica.” Accessed: Feb. 01, 2024. [Online]. Available: https://www.britannica.com/science/plant-disease

[3] R. K. Horst, “Plant,” *Westcott’s Plant Disease Handbook*, pp. 65–530, 2001.

[4] B. J. Cardinale *et al.*, “The functional role of producer diversity in ecosystems,” *American J*

of Botany, vol. 98, no. 3, pp. 572–592, Mar. 2011, doi: 10.3732/ajb.1000364.

[5] Y. Fang and R. Ramasamy, "Current and Prospective Methods for Plant Disease Detection," *Biosensors*, vol. 5, no. 3, pp. 537–561, Aug. 2015, doi: 10.3390/bios5030537.

[6] A. Y. Khaled, S. Abd Aziz, S. K. Bejo, N. M. Nawi, I. A. Seman, and D. I. Onwude, "Early detection of diseases in plant tissue using spectroscopy – applications and limitations," *Applied Spectroscopy Reviews*, vol. 53, no. 1, pp. 36–64, Jan. 2018, doi: 10.1080/05704928.2017.1352510.

[7] F. Martinelli *et al.*, "Advanced methods of plant disease detection. A review," *Agron. Sustain. Dev.*, vol. 35, no. 1, pp. 1–25, Jan. 2015, doi: 10.1007/s13593-014-0246-1.

[8] C. Sarkar, D. Gupta, U. Gupta, and B. B. Hazarika, "Leaf disease detection using machine learning and deep learning: Review and challenges," *Applied Soft Computing*, vol. 145, p. 110534, Sep. 2023, doi: 10.1016/j.asoc.2023.110534.

[9] A. A. Soofi and A. Awan, "Classification techniques in machine learning: applications and issues," *Journal of Basic & Applied Sciences*, vol. 13, no. 1, pp. 459–465, 2017.

[10] M. Lamba, Y. Gigras, and A. Dhull, "Classification of plant diseases using machine and deep learning," *Open Computer Science*, vol. 11, no. 1, pp. 491–508, Dec. 2021, doi: 10.1515/comp-2020-0122.

[11] K. M. Hosny, W. M. El-Hady, F. M. Samy, E. Vrochidou, and G. A. Papakostas, "Multi-Class Classification of Plant Leaf Diseases Using Feature Fusion of Deep Convolutional Neural Network and Local Binary Pattern," *IEEE Access*, vol. 11, pp. 62307–62317, 2023, doi: 10.1109/ACCESS.2023.3286730.

[12] B. Li, J. Tang, Y. Zhang, and X. Xie, "Ensemble of the Deep Convolutional Network for Multiclass of Plant Disease Classification Using Leaf Images," *Int. J. Patt. Recogn. Artif. Intell.*, vol. 36, no. 04, p. 2250016, Mar. 2022, doi: 10.1142/S0218001422500161.

[13] S. Jadhav and A. M. Lal, "Multi-Class Plant Leaf Disease Detection Using a Deep Convolutional Neural Network:," *International Journal of Information System Modeling and Design*, vol. 13, no. 1, pp. 1–14, Dec. 2022, doi: 10.4018/IJISMD.315126.

[14] J. Hang, D. Zhang, P. Chen, J. Zhang, and B. Wang, "Classification of Plant Leaf Diseases Based on Improved Convolutional Neural Network," *SENSORS*, vol. 19, no. 19, Oct. 2019, doi: 10.3390/s19194161.

[15] V. Tiwari, R. C. Joshi, and M. K. Dutta, "Deep neural network for multi-class classification of medicinal plant leaves," *Expert Systems*, vol. 39, no. 8, p. e13041, Sep. 2022, doi: 10.1111/exsy.13041.

[16] M. A. Noyan, "Uncovering bias in the PlantVillage dataset," 2022, doi: 10.48550/ARXIV.2206.04374.

[17] T. O. EMMANUEL, "PlantVillage Dataset." Accessed: Aug. 23, 2023. [Online]. Available: https://www.kaggle.com/datasets/emmarex/plantdisease

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT: IEEE, Jun. 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[19] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL: IEEE, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[20] X. Huang *et al.*, "Tomato Leaf Disease Detection System Based on FC-SNDPN," *Multimed Tools Appl*, vol. 82, no. 2, pp. 2121–2144, Jan. 2023, doi: 10.1007/s11042-021-11790-3.

[21] J. Yao, S. N. Tran, S. Sawyer, and S. Garg, "Machine learning for leaf disease classification: data, techniques and applications," *Artif Intell Rev*, vol. 56, no. S3, pp. 3571–3616, Dec. 2023, doi: 10.1007/s10462-023-10610-4.

[22] S. Jha, V. Luhach, G. S. Gupta, and B. Singh, "Crop Disease Classification using Support Vector Machines with Green Chromatic Coordinate (GCC) and Attention based feature extraction for IoT based Smart Agricultural Applications," Synthical.

Accessed: Feb. 02, 2024. [Online]. Available: https://synthical.com/article/7cf24688-ccad-4bf0-a8db-48deb3ff04f5

[23] R. Chen, H. Qi, Y. Liang, and M. Yang, "Identification of plant leaf diseases by deep learning based on channel attention and channel pruning," *Front Plant Sci*, vol. 13, p. 1023515, 2022, doi: 10.3389/fpls.2022.1023515.

[24] A. Halim, C. Chow, M. Amabel, S. Achmad, and R. Sutoyo, "The Impact of Hyperparameter Tuning in Convolutional Neural Network on Image Classification Model: A Case Study of Plant Disease Detection," in *2023 5th International Conference on Cybernetics and Intelligent System (ICORIS)*, Pangkalpinang, Indonesia: IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/ICORIS60118.2023.10352209.