Plant Disease Detection and Classification using a Deep learning-based framework

Mridul Ghosh[1], Asifuzzaman Lasker[2], Poushali Banerjee[3], Anindita Manna[3],Sk Md Obaidullah[2], Teresa Gonçalves[4], Kaushik Roy [5]

1 Dept. of Computer Science, ShyampurSiddheswariMahavidyalaya; mridulxyz@gmail.com

2 Dept. of Computer Science & Engineering, Aliah University

3 Dept. of Computer Science, Surendranath College; poushalib30@gmail.com, mannaanindita2018@gmail.com

4 Departamento de Informática, Universidade de Évora

5 Dept. of Computer Science, West Bengal State University * Correspondence: mridulxyz@gmail.com;

Abstract. Plant diseases pose a significant threat to agriculture, causing substantial yield losses and economic damages worldwide. Traditional methods for detecting plant diseases are often time-consuming and require expert knowledge. In recent years, deep learning-based approaches have demonstrated great potential in the detection and classification of plant diseases. In this paper, we propose a Convolutional Neural Network (CNN) based framework for identifying 15 categories of plant leaf diseases, focusing on Tomato, Potato, and Bell pepper as the target plants. For our experiments, we utilized the publicly available PlantVillage dataset. The choice of a CNN for this task is justified by its recognition as one of the most popular and effective deep learning methods, especially for processing spatial data like images of plant leaves. We evaluated the performance of our model using various performance metrics, including accuracy, precision, recall, and F1-score. Our findings indicate that our approach outperforms state-of-the-art techniques, yielding encouraging results in terms of disease identification accuracy and classification precision.

Keywords: Plant disease; Deep learning; CNN; Leaf disease classification.

1 Introduction

Plant diseases pose significant threats to agricultural productivity, food security, and the global economy [1]. Detecting and diagnosing plant diseases at an early stage is crucial for effective disease management and prevention. Over the years, advances in technology and the application of various scientific methods have greatly improved the process of plant disease detection. The detection involves identifying and determining the presence of pathogens, such as bacteria, fungi, viruses, and other harmful microorganisms, that can cause diseases in plants. Timely and accurate detection enables farmers, plant pathologists, and researchers to take appropriate measures to mitigate the spread and impact of diseases, thus minimizing crop losses and ensuring sustainable agricultural practices.

Traditional methods of plant disease detection primarily relied on visual observations of symptoms exhibited by the infected plants. These symptoms could include wilting, discoloration, leaf spots, abnormal growth patterns, and various other physical changes. While visual inspection remains a valuable tool, it is often limited by the subjectivity of human observation and the difficulty in differentiating between similar symptoms caused by different pathogens. In recent years, technological advancements have revolutionized plant disease detection by providing more precise, rapid, and reliable methods. Here are some of the key techniques and tools used in modern plant disease detection:

1. Molecular Techniques: Polymerase Chain Reaction (PCR), DNA sequencing, and other molecular biology methods [2] are widely employed for the identification and characterization of plant pathogens. These techniques enable the detection of specific DNA or RNA sequences unique to a particular pathogen, allowing for highly accurate and targeted diagnosis.
2. Immunoassays: Enzyme-linked immunosorbent Assays (ELISA) and other immunological techniques [3] are utilized to detect the presence of plant pathogens based on the specific immune response generated by the host plants. These tests rely on the recognition and binding of pathogen-specific antigens by antibodies, providing a sensitive and specific detection method.
3. Machine learning (ML) and Artificial Intelligence (AI): By leveraging ML and AI-based models [6,7,8], large datasets of plant images, genomic sequences, and environmental parameters can be analyzed to develop predictive models for disease detection. These models can identify patterns, correlations, and anomalies that may not be apparent to human observers, improving the accuracy and efficiency of detection.

By using machine learning-based work, the performance of classifying the diseases obtained is high and the time of classification is much less in comparison with other techniques. For this, in this work, a deep learning-based [9-11] framework is designed to classify the Tomato and Potato leaf diseases. Here, a CNN [12, 13] based model is developed for the classification of Tomato, Potato, and Bell pepper plant diseases.

Kaur et al. [14] employed a CNN-based framework to detect diseases in tomato, potato, and grape leaves. They utilized a dataset containing more than 4,000 images of diseased and healthy leaves and trained several CNN models to classify the images as either diseased or healthy. The best-performing model achieved an accuracy of 98.5%. In [15], the authors proposed a technique that involves acquiring images of healthy and diseased tomato plants, pre-processing the images to remove noise and unwanted regions, extracting features from the pre-processed images, selecting relevant features, and training an SVM classifier to categorize the tomato images as healthy or diseased. The proposed technique was evaluated on a dataset of tomato images with four diseases: Bacterial Spot, Early Blight, Late Blight, and Septoria Leaf Spot. The results demonstrated an accuracy of 94.7% in detecting tomato diseases using the proposed technique. Tiwari et al. [14] developed an automated system to diagnose and classify diseases like early blight, late blight, and healthy conditions in potato leaves, offering a novel solution. The results demonstrated an accuracy of 97.8% over the test dataset, with improvements of 5.8% and 2.8%. Srinivasan et al. [15] proposed an image categorization technique to identify healthy and unhealthy leaves from a multilevel image dataset. Additionally, it identifies the specific type of disease affecting the unhealthy leaves. Using the CNN technique, they extracted 39 types of diseases in 13 crop species from the PlantVillage image dataset, achieving an accuracy of 98.75% at epoch 25.

In Section 2 the materials and method are discussed, followed by the experiment and results analysis in Section 3. Finally, we present our conclusion and future work in Section 4.

2 Material & Method

Deep learning models based on CNN frameworks [9,10] are specifically designed for processing and analyzing visual data, such as images. These models utilize layers that play a vital role in extracting significant features and patterns from raw input images. Here are some key CNN layers:

i. Convolutional Layer: The convolutional layer applies a set of filters or kernels to the input image, performing a mathematical operation known as convolution. This operation detects local patterns by sliding the filters across the image and calculating dot products between filter weights and pixel values. Convolutional layers are capable of capturing features like edges, textures, and shapes.

ii. Pooling Layer: Pooling layers reduce the spatial dimensions of the feature maps obtained from convolutional layers. The most common pooling operation is max pooling, which downsamples the input by selecting the maximum value within a predefined neighbourhood. Pooling helps to achieve spatial invariance and reduces the computational burden by summarizing the most salient features.

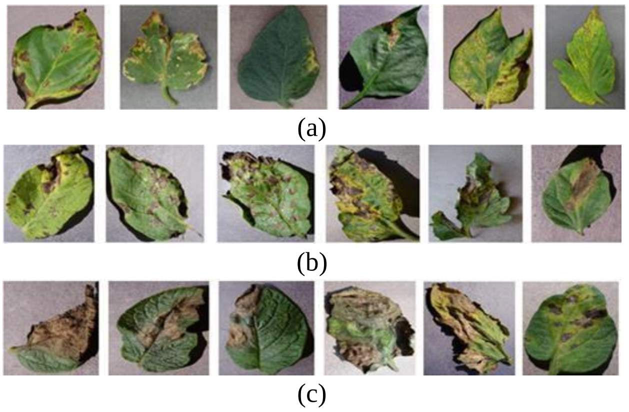
iii. Fully Connected/dense Layer: The fully connected layer is typically placed after the convolutional and pooling layers. It connects all neurons from the previous layer to every neuron in the current layer. It captures high-level abstractions by combining the features learned from previous layers. The output of this layer is often fed into a Soft-Max activation function for classification.

2.1 Dataset

In this study, we utilized the publicly available PlantVillage dataset, curated by Hughes & Salathe [16] (https://www.kaggle.com/datasets/emmarex/plantdisease), for the purpose of detecting plant leaf diseases. The PlantVillage dataset comprises a total of 20,639 images, thoughtfully organized into two categories: diseased and healthy leaves of various plant species. These images have been expertly classified by specialists in the field of plant pathology. The dataset primarily focuses on three plant species: Bell Pepper, Potato, and Tomato. The photographs were taken with leaves positioned against a paper sheet, which provided a consistent gray or black background. Within the PlantVillage dataset, there are 15 distinct classes of diseases, as listed in Table 1. For visual reference, a selection of sample images from the dataset can be observed in Fig. 1.

Table 1. The details specification of Plant Village dataset [16].

Plant names	Disease names	No. of images
Bell pepper	Healthy	1478
	Diseased: Bacterial spot	997
Potato	Healthy	152
	Diseased: Early Blight	1000
	Diseased: Late Blight	1000
Tomato	Healthy	1591
	Diseased: Mosaic Virus	373
	Diseased: Bacterial spot	2127
	Diseased: Early Blight	1000
	Diseased: Late Blight	1909
	Diseased: Leaf Mold	952
	Diseased: Septoria Leaf Spot	1771
	Diseased: Two Spotted Spider mites	1676
	Diseased: Target Spot	1404
	Diseased: Yellow Leaf Curl Virus	3209



(a)

(b)

(c)

Fig. 1. Sample leaf images of affected plants: in (a) images of Bacterial Spot disease, in Bellpepper (b) Early Blight, in Potato and (c) Late Blight Tomato.

3.2 Data Pre-Processing:

The pixel values of the images are rescaled to be in the range of 0 to 1 by applying the rescale argument with a value of 1/255. This step ensures that the pixel values are normalized, which can help improve the training process. The dataset is split into training and validation sets, this split allows for evaluating the model's performance on unseen data. This resizing ensures that all input images have consistent dimensions. It helps in standardizing the input size for the model. Here, the images are resized to a target size of (224, 224). These pre-processing steps ensure that the input images are appropriately rescaled and resized before being fed into the model for training and evaluation.

3.3 Proposed Model:

Plants are vulnerable to a range of disease-related disorders and infections. These issues can arise from various causes, including disturbances caused by environmental conditions such as temperature, humidity, inadequate or excessive nutrients, and light exposure. Additionally, plants commonly face bacterial, viral, and fungal diseases.

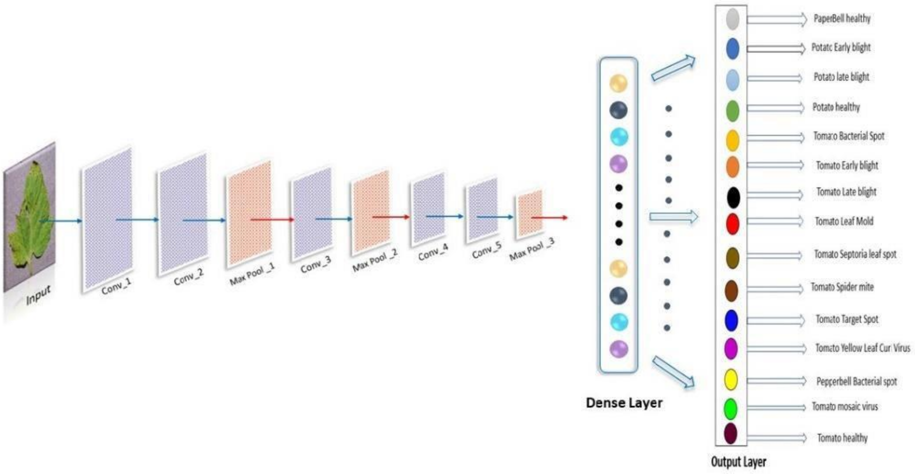


Fig. 2. The proposed C5P3D2 architecture for classification of the plant diseases

To classify the fifteen categories of potato and tomato plant disease, we propose a plant disease classification architecture called C5P3D2 which is shown in Fig. 2. This architecture consists of 5 Convolutional layers, 3 max-pooling layers, and 2 dense layers. The input image is passed through the first Convolutional layer with 128 filters of size 5x5. The resulting feature map is then fed into another Convolutional layer with a filter size of 64 of size 3x3. The output of this Convolutional layer is further processed by a max-pooling layer of filter size 2x2.The output of this pooling layer is further processed by another Convolutional layer with 32 filters of size 3x3. After this the resulting feature map is then fed into another max-pooling layer of filter size 2x2. Next, the output from the previous layer is inputted into a pair of Convolutional layers with filter sizes of 3x3 and 16, and 3x3 and 16, respectively. Each Convolutional layer captures different aspects of the input features. After each Convolutional layer, a max-pooling layer with a filter size of 3x3 is applied. Two batch normalization layers are utilized before the dense 1 and dense2 to enhance the stability and convergence of the network. Subsequently, dropout regularization is employed with dropout rates of 0.5 after the dense layers. Two dense layers are included in the architecture with sizes of 128 and 15, respectively. These layers perform weighted sums of the input features. Finally, the output layer consists of 15 neurons, representing the classification categories. The generated parameters in our model are tabulated in Table 2.

Table 2. The generated parameters in C5P3D2 model

Layers	Output dimension	#Parameter
Convolution_1	(220,220,128)	9,728
Convolution_2	(218,218,64)	73,792
Convolution_3	(107,107,52)	18,464
Convolution_4	(51,51,16)	4,624
Convolution_5	(49,49,16)	2,320
Batchnormalization_1	9216	36,864
Dense_1	128	11,79,776
Batchnormalization_2	128	512
Dense_2	15	1,935
Total Parameters	1,328,015	
Total Trainable Parameters	1,309,327	

3 Experiment & results

3.1 System setup

Our proposed system was tested and assessed on a GPU machine equipped with dual Intel(R) Core(TM) i5-10400H CPUs, operating at a clock speed of 2.60GHz, accompanied by 32 GB of RAM, and powered by two NVIDIA Quadro RTX 5000 GPUs. The machine ran on Windows 10 Pro OS version 20H2 and utilized TensorFlow 2.0.0 for training and inference of deep learning models.

3.2 Training regime

The dataset is split into train and test sets using different ratios. Dataset split into train-test sets in CNN training to evaluate model performance on unseen data, prevent overfitting, and optimize hyperparameters. Training set used for learning, test set for unbiased evaluation. Specifically, the ratios used are 8:2, 7:3, and 6:4. These ratios determine the proportion of images allocated to the training and testing sets. For the 8:2 ratios, out of the total image pool of 20, 639, 16, 512 images are assigned to the training set, while 4, 127 images are allocated to the testing set. For the 7:3 ratios, 14,444 images are used for training, and 6,192 images are used for testing. For the 6:4 ratios, 12,383 images are used for training, and 8,256 images are used for testing. The training and testing sets are used independently of each other to avoid data leakage. This means that the models are trained only on the training set and evaluated only on the testing set.

3.3 Evaluation Protocols

There are four different evaluation protocols are considered in this work. Apart from accuracy precision, recall, and F1Score are measured to evaluate the model's performance. The Equations of the corresponding evaluation parameters are presented below:

$$\text{Accuracy} = (TruePositives + TrueNegatives)/\ (TruePositives + TrueNegatives + FalsePositives + FalseNegatives) \qquad (1)$$

$$\text{Precision} = TruePositives\ /\ (TruePositives + FalsePositives) \qquad (2)$$

$$\text{Recall} = TruePositives/\ (TruePositives + FalseNegatives) \qquad (3)$$

$$\text{F1 score}= 2^*\ (precision*recall/\ precision\ +recal) \qquad (4)$$

In our C5P3D2 architecture, Categorical Cross-Entropy Loss function was used which can be described by Equation 5.

$$L=-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}) \qquad (5)$$

Here, N is the number of samples in the batch, is the number of classes in the classification problem, $y_{ij}$is an indicator function that is 1 if the sample i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that sample i belongs to class j.

3.4 Result & discussions

3.4.1 Ablation Study

Table 3. Ablation study to build a CNN-based framework to get higher performance

Architecture	Convolutional	Pooling	Dense	Accuracy (%)
CPD	5x5	3x3	256	81.54
CCPPDD	5x5, 3x3	3x3, 2x2	256,128	84.02
CPPDD	3x3	2x2, 2x2	256, 128	85.05
CCPDD	5x5, 3x3	3x3	256, 128	79.52
CCPPD	3x3, 5x5	3x3, 2x2	256	86.35
C5P3D2	5x5,3x3,3x3,3x3,3x3	2x2, 3x3, 2x2	128, 15	94.95

We performed a rigorous experiment to establish a suitable framework for our problem. We have used different combinations of convolutional, pooling, and dense layers along with the different dimensional filters. In Table 3 the outcome of these experiments is tabulated considering 8:2 train-test set, batch size of 32, and 100 epoch trial runs. Finally, C5P3D2 architecture was considered for our work.

3.4.2 Our results

In Table 4 the performance of our C5P3D2 architecture was measured by changing the epoch values of 20, 50, and 100 for different train-test ratios and batch size of 32 at the beginning. The maximum accuracy of 94.95% has been obtained by running 100 epochs in 80:20 ratios. Considering 100 epochs, the rest of the experiments were performed. In Fig. 3 and 4 represents the training performance of C5P3D2 architecture with training and validation curve.

Table 4. The accuracies for different train-test ratios for 20, 50 and 100 epochs run

Train-test Ratio	Epoch	Accuracy (%)
80:20	20	77.58
	50	94.00
	100	94.95
70:20	20	93.76
	50	94.45
	100	94.62
60:40	20	91.85
	50	92.43
	100	94.13

From Table 5 it is observed that in 100 epochs the highest accuracy and precision are obtained 94.95% and 94.94%, respectively. So, we keep this epoch constant for the rest of our experiment. Considering the epoch value as 100 and batch size 32, different train-test ratios were considered. We further investigated by changing the batch size of 32, 64, and 128 considering the 100 epochs and 80:20 train-test ratio which is tabulated in Table 6. It is seen that there are improvements in accuracy, precision, recall, and f-score, respectively. The train-test accuracy and loss curves for 100 epochs 128 batch size and 80:20 train test ratio are shown in Figs. 3 (a) and (b), respectively.

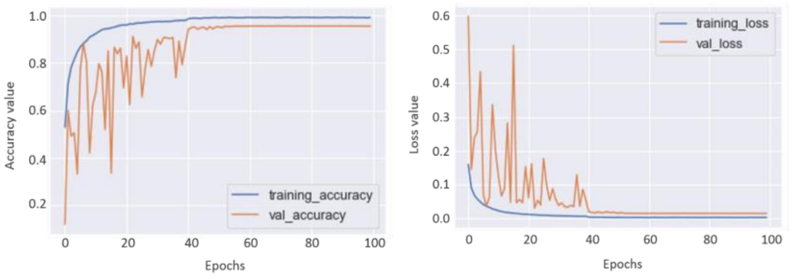
Table 5. For the 8:2 train-test ratio the performance measurement by changing the epochs

pochs	Accuracy	Precision	Recall	F1-Score
20	77.85%	82.35%	77.58%	75.07%

50	94%	94%	94%	93%
100	94.95%	94.94%	94.95%	94.91%

Table 6. For 100 epochs, 8:2 train-test ratio for different batch size

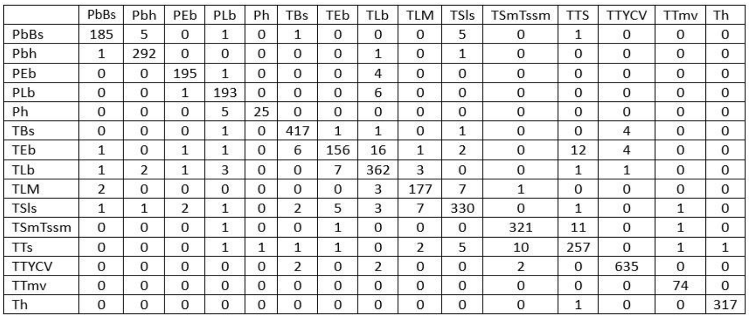
Batch size	Accuracy	Precision	Recall	F1-score
32	94.95%	94.94%	94.95%	94.91%
64	95.41%	95.43%	95.41%	94.46%
128	**95.51%**	**95.51%**	**95.24%**	**95.47%**



(a) (b)

Fig. 3. Training and validation accuracy (a) and loss curves (b) for 100 epochs 128 batch size and 80:20 train test ratio.

Table 7. The confusion matrix of the highest result i.e., corresponding to the accuracy of 95.51% for 128 batch size in 100 epochs of our proposed model is presented.



	PbBs	Pbh	PEb	PLb	Ph	TBs	TEb	TLb	TLM	TSls	TSmTssm	TTS	TTYCV	TTmv	Th
PbBs	185	5	0	1	0	1	0	0	0	5	0	1	0	0	0
Pbh	1	292	0	0	0	0	0	1	0	1	0	0	0	0	0
PEb	0	0	195	1	0	0	0	4	0	0	0	0	0	0	0
PLb	0	0	1	193	0	0	0	6	0	0	0	0	0	0	0
Ph	0	0	0	5	25	0	0	0	0	0	0	0	0	0	0
TBs	0	0	0	1	0	417	1	1	0	1	0	0	4	0	0
TEb	1	0	1	1	0	6	156	16	1	2	0	12	4	0	0
TLb	1	2	1	3	0	0	7	362	3	0	0	1	1	0	0
TLM	2	0	0	0	0	0	0	3	177	7	1	0	0	0	0
TSls	1	1	2	1	0	2	5	3	7	330	0	1	0	1	0
TSmTssm	0	0	0	1	0	0	1	0	0	0	321	11	0	1	0
TTs	0	0	0	1	1	1	1	0	2	5	10	257	0	1	1
TTYCV	0	0	0	0	0	2	0	2	0	0	2	0	635	0	0
TTmv	0	0	0	0	0	0	0	0	0	0	0	0	0	74	0
Th	0	0	0	0	0	0	0	0	0	0	0	1	0	0	317

Here full names corresponding to the acronyms are:

PbBs=Peeper_bell_Bacterial_spot, Pbh=Peeper_bell_healthy,
PEb=Potato_Early_blight, PLb=Poato_Late_blight, Ph=Potato_healthy,

TBs=Tomato_Bacterial_spot, TEb=Tomato_Early_blight,
TLb=Tomato_Late_blight,TLM=Tomato_Leaf_Mold,
Tls=Tomato_Septoria_leaf_spot,
TSmTssm=Tomato_Spider_mites_Two_spotted_spider_mites,
TTs=Tomato_Target_Spot,
TTYCV= Tomato_Tomato_YellowLeaf_Curl_Virus,
TTmv= Tomato_Tomato_mosaic_virus,
Th= Tomato_healthy.

It is observed from the confusion matrix shown in (Table 7). that for the class PbBs, 185 instances are correctly classified, 292 instances are correctly classified for Pbh,, 195 instances are correctly classified for Potato_Eb, 193 instances are correctly classified for PLb, 25 instances are correctly classified for Ph, 417 instances are correctly classified for TBs, 156 instances are correctly classified for TEB, 362 instances are correctly classified for TLb, 177 instances are correctly classified for TLM, 330 instances are correctly classified for TSls, 321 instances are correctly classified for TSmTssm, 257 instances are correctly classified for TTS, 635 instances are correctly classified for TTYCV, 74 instances are correctly classified for TTmv and 317 instances are correctly classified.

3.4.3 Testing the model with external data

In the testing phase, we evaluated external images, and as shown in Fig. 4, the model accurately classified a "Pepper bell healthy" image. This classification aligns perfectly with the ground truth observation, as expected.

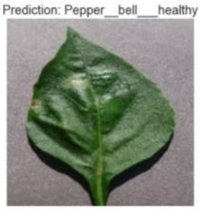


Fig. 4. Pepper bell healthy image tested using C5P3D2 architecture.

3.4.4 Comparison

Our C5P3D2-based approach for plant disease detection demonstrates competitive results compared to other studies. Which is presented in Table 8. While Prajwala TM et al. [17] achieved 94-95% accuracy with a LeNet variant for tomato leaves, our model, which extends to multiple plants, shows a gain in accuracy of more than 1%. Salih et al.'s [18] CNN model yielded a gain in accuracy of less than 1%, despite using several levels of deep learning approaches. Basavaiah and Anthony's [19] multi-feature fusion method did not achieve a significant improvement in accuracy compared to our approach. It is seen that our method produced 18.5% higher accuracy compared to the MobileNetV4.

Table 8. Comparison of our method with standard CNN on PlantVillage dataset

Methods	Accuracy	Recall	Precision	F1-Score
VGG-16	52.50%	51.50%	50.02%	52.49%
MobileNetV3	71.40%	56.10%	70.41%	82.43%
MobileNetV4	77.01%	66.05%	65.02%	66.50%
Proposed	95.51%	95.51%	94.24%	95.47%

4 Conclusion & Future Scope

We have designed a C5P3D2 framework for the detection of plant diseases in pepper bell, tomato, and potato leaves. The system successfully takes image inputs from the user and provides output indicating the detected disease. This enables farmers to take appropriate preventive measures and use the correct pesticides. This proposed framework can be extended to other crops that suffer from diseases, given the availability of a sufficiently large dataset for that specific crop. In the future, we will consider improving the performance of the method by utilizing more advanced deep learning-based techniques. We plan to deploy the system as a GUI-based platform. The web interface may also include a forum for farmers to discuss the current trends they face in different diseases. The future of plant disease detection using CNNs involves expanding and di-versifying training datasets, optimizing CNN architectures for enhanced performance, integrating real-time monitoring technologies, exploring multimodal analysis techniques, improving the interpretability of CNN models, and addressing practical challenges for widespread deployment in agricultural settings.

References

1. Anand, G., &Rajeshkumar, K. C. (2022). Challenges and Threats Posed by Plant Pathogenic Fungi on Agricultural Productivity and Economy. In Fungal diversity, ecology and control management (pp. 483-493). Singapore: Springer Nature Singapore.
2. Joshi, M., & Deshpande, J. D. (2010). Polymerase chain reaction: methods, principles and application. International Journal of Biomedical Research, 2(1), 81-97.
3. Tijssen, P., & Adam, A. (1991). Enzyme-linked immunosorbent assays and developments in techniques using latex beads. Current Opinion in Immunology, 3(2), 233-237.
4. Flores, A. M., Demsas, F., Leeper, N. J., & Ross, E. G. (2021). Leveraging machine learning and artificial intelligence to improve peripheral artery disease detection, treatment, and outcomes. Circulation research, 128(12), 1833-1850.

5. Ghosh, M., Obaidullah, S. M., Gherardini, F., &Zdimalova, M. (2021). Classification of geometric forms in mosaics using deep neural network. Journal of Imaging, 7(8), 149.
6. Ghosh, M., Mukherjee, H., Obaidullah, S. M., & Roy, K. (2021). STDNet: A CNN-based approach to single-/mixed-script detection. Innovations in Systems and Software Engineering, 17(3), 277-288.
7. Ghosh, M., Roy, S. S., Mukherjee, H., Obaidullah, S. M., Santosh, K. C., & Roy, K. (2022). Understanding movie poster: transfer-deep learning approach for graphic-rich text recognition. The Visual Computer, 1-20.
8. Ghosh, M., Mukherjee, H., Obaidullah, S. M., Santosh, K. C., Das, N., & Roy, K. (2021). LWSINet: A deep learning-based approach towards video script identification. Multimedia Tools and Applications, 80(19), 29095-29128.
9. Ghosh, M., Roy, S. S., Mukherjee, H., Obaidullah, S. M., Gao, X. Z., & Roy, K. (2021). Movie title extraction and script separation using shallow convolution neural network. IEEE Access, 9, 125184-125201.
10. Lasker, A., Ghosh, M., Obaidullah, S. M., Chakraborty, C., & Roy, K. (2023). LWSNet-a novel deep-learning architecture to segregate Covid-19 and pneumonia from x-ray imagery. Multimedia Tools and Applications, 82(14), 21801-21823.
11. Lasker, A., Obaidullah, S. M., Chakraborty, C., & Roy, K. (2022). Application of machine learning and deep learning techniques for Covid-19 screening using radiological imaging: a comprehensive review. SN computer science, 4(1), 65.
12. Kaur, S., Pandey, S., & Goel, S. (2019). Plants disease identification and classification through leaf images: A survey. Archives of Computational Methods in Engineering, 26, 507-530.
13. Verma, S., Chug, A., & Singh, A. P. (2018, September). Prediction models for identification and diagnosis of tomato plant diseases. In 2018 International Conference on advances in computing, communications and informatics (ICACCI) (pp. 1557-1563). IEEE..
14. Tiwari, D., Ashish, M., Gangwar, N., Sharma, A., Patel, S., & Bhardwaj, S. (2020, May). Potato leaf diseases detection using deep learning. In 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 461-466). IEEE.
15. Srinivasan, R., Santhanakrishnan, C., Iniyan, S., Subash, R., &Sudhakaran, P. (2023). CNN-based Plant Disease Identification in Crops from Multilabel Images using Contextual Regularization. Journal of Survey in Fisheries Sciences, 10(2S), 522-531.
16. Hughes, D., &Salathé, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060.
17. Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., &Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In 2018 eleventh int. conf. on contemporary computing (IC3) (pp. 1-5). IEEE.
18. Salih, T. A. (2020). Deep learning convolution neural network to detect and classify tomato plant leaf diseases. Open Access Library Journal, 7(05), 1.
19. Basavaiah, J., & Arlene Anthony, A. (2020). Tomato leaf disease classification using multiple feature extraction techniques. Wireless Personal Communications, 115(1), 633-651.