IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

SMART AGRICULTURE: LEVERAGING CONVOLUTIONAL NEURAL NETWORKS FOR PLANT DISEASE DETECTION

[1]C.Manogari, [2] Dr. R. Senthamil Selvi
[1] M.E Student,Department of CSE,Saranathan College of Engineering, Trichy,India.
[2] Professor, Department of CSE, Saranathan college of Engineering, Trichy,India.

Abstract: Advancements in smart agriculture have paved the way for the integration of cutting-edge technologies in traditional farming practices. This study explores the application of Convolutional Neural Networks (CNNs) for early and accurate detection of plant diseases, a critical factor for sustainable crop management. By analyzing leaf images, the CNN model identifies disease symptoms with high precision, enabling timely interventions. The proposed approach demonstrates significant potential to reduce agricultural losses, enhance crop yields, and promote eco-friendly farming practices. This research underscores the transformative role of AI-driven solutions in modern agriculture.

IndexTerms**:** Convolutional Neural Networks (CNN),Transfer Learning, Deep Learning, Plant Leaf Diseases, Image Processing,Smart agriculture, Accuracy, Precision, Recall.

1 INTRODUCTION

Agriculture is a vital part of the global economy, supporting food production essential for human survival. Plant diseases caused by biotic and abiotic factors can significantly impact crop quality and yield, especially in countries dependent on agriculture for income and employment. Early detection and treatment of plant diseases are crucial to minimize losses and optimize harvests. Advances in computer vision, machine learning, and deep learning, including Convolutional Neural Networks (CNNs), have improved plant disease identification. However, challenges remain, such as handling redundant datasets, small datasets, and complex models requiring extensive training. Research should focus on efficient datasets and augmentation techniques for better classification, addressing these limitations to enhance plant disease detection and management.

However, extant studies employing deep learning techniques encounter challenges such as managing large datasets with redundant information, small datasets representing single disease types, and the deployment of complex models necessitating extensive training time and high computational costs. To surmount these obstacles, research should prioritize the use of compact, efficient datasets and the application of augmentation techniques for more precise classification. Our study endeavors to address these gaps by proposing a more efficient and effective approach to plant disease classification, leveraging deep learning methodologies prevalent in contemporary research.

1.1 Related Work

Classifying plant diseases is challenging for farmers globally. Traditional methods are labor-intensive and time-consuming, impacting crop productivity. Early detection is crucial to prevent outbreaks. This section reviews current research on plant disease classification.Faiqua Adnan et al. [2] introduced a deep learning model-AADLfor precise disease identification of plant leaf diseases , covering 52 classes. The model were trained together to improve classification accuracy, achieving a 98.71% accuracy rate.Yang Liu et al. [6] developed a neural network architecture called SqueezeNet for disease classification. They utilized rectified linear units (ReLU) in place of the tanh function in AlexNet, resulting in an execution speed six times faster than a CNN using the tanh activation function. The model achieved an overall accuracy of 91.94%.Sanasam Premananda Singhet al. [17] introduced an improved CNNmodel based on the CNN framework. This model included two inception modules, a pooling layer, and a new activation function, resulting in an average accuracy of 99.6% for identifying plant leaf disease.

In summary, the primary challenge for researchers is to develop a deep learning-based framework for plant disease classification that minimizes limitations and yields precise results. The literature review reveals that existing studies face several issues, including the use of diverse datasets with varying sizes and sources, often containing redundant data, and the employment of complex models, which result in longer training times and higher computational costs. These challenges lead to problems such as overfitting, increased computational costs, and a need for enhanced accuracy and efficiency.

1.2 Motivation and Objective

The primary objective of this research endeavor is to accurately identify various plant leaf diseases through effective segmentation techniques. Additionally, this study addresses several existing issues, such as inadequate image pre-processing, low segmentation accuracy, and ineffective disease classification. Agricultural productivity is significantly impacted by diseases affecting plant leaves, including blight, rust, and bacterial infections. Consequently, numerous researchers have contributed to the detection and classification of plant leaf diseases. However, certain research gaps remain as, Inadequate Image Pre-treatment, Segmentation Accuracy, Ineffective Classification. Additionally, the improper use of color space model conversion results in minimal accuracy in plant leaf disease detection. By addressing these issues, our research aims to enhance the accuracy and efficiency of plant leaf disease identification and classification. The proposed research aims to significantly improve the accuracy and reliability of the system used for more accurately identifying plant diseases using a deep CNN model and the EfficientNetB0 algorithm.

2 MATERIALS

2.1 Dataset

The dataset utilized in this research is sourced from Kaggle and comprises 3 plant species with 15 classes of both healthy and diseased plants, a total of 20,639 images. Details of each class are shown in Table 1, Table 2, and Table 3. The dataset was refined to include only those classes with a significant sample size, meaning only classes with at least 36 images were retained, while those with fewer were excluded. The data was then split into training and test sets using the train-test split function from the sci-kit learn library. Specifically, 90% of the data was allocated for training, with the remaining 10% for test sets. Each image in the dataset has dimensions of 256 × 256 × 3, indicating a width and height of 256 pixels and three channels representing the RGB format. These images were subsequently resized as per network requirements. Some of the sample images available for each class of plant diseases have been shown in Figure 1.

Table 1 Potato Leaf Dataset

Class	Disease	Type of Disease	No of Images
C1	Early Blight	Fungus	1000
C2	Late Blight	Fungus	1000
C3	Healthy	-	152

Table 2 Bell Pepper leaves dataset

Class	Disease	Type of Disease	No of Images
C1	Bacterial Spot	Bacteria	997
C2	Healthy	-	1478

Table 3 Tomato Leaves Dataset

Class	Disease	Type of Disease	No of Images
C1	Bacterial spot	Bacteria	2127
C2	Early Blight	Fungus	1000
C3	Late Blight	Fungus	1909
C4	Leaf Mold	Fungus	952
C5	Septoria Leaf Spot	Fungus	1771
C6	Spider Mites	Pest	1676
C7	Target Spot	Fungus	1404
C8	Tomato Yellow Leaf Curl	Virus	3209
C9	Tomato Mosaic Virus	Virus	373
C10	Healthy	-	1591

Figure 1-Class-wise sample Images of the Plant Village Dataset

3 METHODOLOGY

3.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are widely regarded as one of the most prominent and versatile algorithms in the realm of deep learning. Their key strength lies in their ability to identify distinguishing features independently, without requiring explicit human input. CNNs have found extensive applications across various fields, such as computer vision, speech processing, facial recognition, image segmentation, image classification, and video analysis, among others. The design of CNNs takes inspiration from the neuronal networks of human and animal brains, resembling traditional neural networks but with a distinctive layered architecture. Typically, a CNN comprises a sequence of convolutional layers followed by subsampling or pooling layers and concludes with fully connected layers.

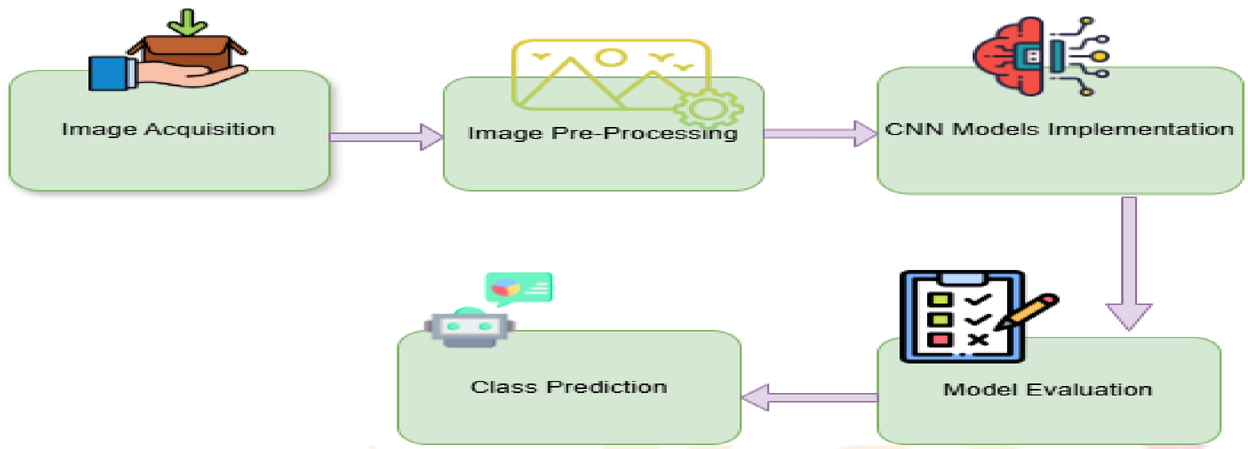


Figure 2 Proposed Framework

These networks process data in a feed-forward manner, ensuring a seamless flow from input to output without recursion. Hidden layers within a CNN include critical components like convolutional, ReLU(Rectified Linear Unit), and Pooling layers, which collectively contribute to its performance and adaptability across tasks. The Rectified Linear Unit (ReLU) is the most prevalent activation function utilized within CNNs, owing to its simplicity and computational efficiency. By converting all negative input values to zero, ReLU introduces non-linearity into the network, enabling it to learn complex patterns and relationships in the data. The function (1) is used to denote ReLUfunction.

$$f(x)_{ReLU} = max(0, x) \qquad (1)$$

ReLU outputs the maximum of zero and the input value, ensuring a sparse activation mechanism. Following the activation step, the outputs from the convolutional layer are processed further. The obtained result is depicted in (2).

$$h^k = f(w^k * x + b^k) \qquad (2)$$

Pooling layers then reduce the spatial dimensions of feature maps by applying operations such as max or average pooling over a localized region of size p×p, significantly diminishing the number of parameters, expediting training, and mitigating overfitting. For classification tasks, CNNs flatten the multidimensional feature maps into a one-dimensional vector, allowing fully connected layers to extract global features and predict class probabilities. The SoftMax activation function (3), applied at the final layer, transforms the outputs into normalized probabilities, representing the likelihood of each class in the dataset.

$$\overrightarrow{\sigma_{(z)}}_i = \frac{e^{\dot{z}_i}}{\sum_{j=1}^{\bar{k}} e^{z_i}} \qquad (3)$$

The SoftMax function takes in a list of numbers, called the input vector, where each number can be any value. The function adjusts these numbers so that they add up to 1, turning them into probabilities. In the dataset, k stands for the total number of possible classes.

3.2 Fine Tuned CNN

The customized CNN model processes the images that have been pre-processed and resized to 256×256 pixels with three color channels. Its architecture is intricately designed to balance high classification accuracy with computational efficiency represented in Figure 3. It includes four convolutional layers, each using a 2×2 kernel with "same" padding to extract essential features from the input images. Following each convolutional layer is a max pooling layer with a 2×2 kernel to down-sample the feature maps, thereby reducing spatial The customized CNN model processes the images that have been pre-processed and resized to 256×256 pixels with three color channels. Its architecture is intricately designed to balance high classification accuracy with computational efficiency represented in Figure 3. It includes four convolutional layers, each using a 2×2 kernel with "same" padding to extract essential features from the input images. Following each convolutional layer is a max pooling layer with a 2×2 kernel to down-sample the feature maps, thereby reducing spatial dimensions while retaining key features. Additionally, the model incorporates three dropout layers, each with a 20% dropout rate, to randomly deactivate neurons during training and prevent overfitting. The network also features a fully connected layer that leads to a dense output layer utilizing the SoftMax activation function for multiclass classification. The weights of the custom CNN model are initialized using the Xavier Glorot uniform method, ensuring uniform distribution within a specified range.This initialization technique avoids the pitfalls of vanishing and exploding gradients, promoting a smooth and stable gradient flow during the backpropagation process. To infuse non-linearity and speed up the learning process, the Rectified Linear Unit (ReLU) activation function is applied across all hidden layers. The Adam optimizer, which stands for adaptive moment estimation, is employed to reduce prediction error and optimize the weights of the nodes. This optimizer merges the benefits of gradient descent with momentum and RMSProp techniques. It dynamically adjusts the learning rate for each weight in the network, enhancing training speed, lowering memory The Adam optimizer, which stands for adaptive moment estimation, is employed to reduce prediction error and optimize the weights of the nodes. This optimizer merges the benefits of gradient descent with momentum and RMSProp techniques. It dynamically adjusts the learning rate for each weight in the network, enhancing training speed, lowering memory usage, and minimizing the need for extensive hyperparameter tuning. For this study, the learning rate is set to 0.001, offering an optimal balance between convergence rate and model accuracy during the training phase. The Categorical Cross-Entropy loss function is used to measure discrepancies between true labels and predicted probabilities. By merging the SoftMax activation function with Cross-Entropy loss, it ensures that class probabilities sum up to 1, penalizing significant deviations from true labels and fostering improved prediction accuracy. The training process spans 10 epochs with a batch size of 32, meaning the model processes 32 images simultaneously before updating the weights. This batch size was selected based on an ablation study to balance training duration and model performance. The callback function is utilized to track validation loss, ensuring that the most effective model is saved as the final version depicted in Figure 3.

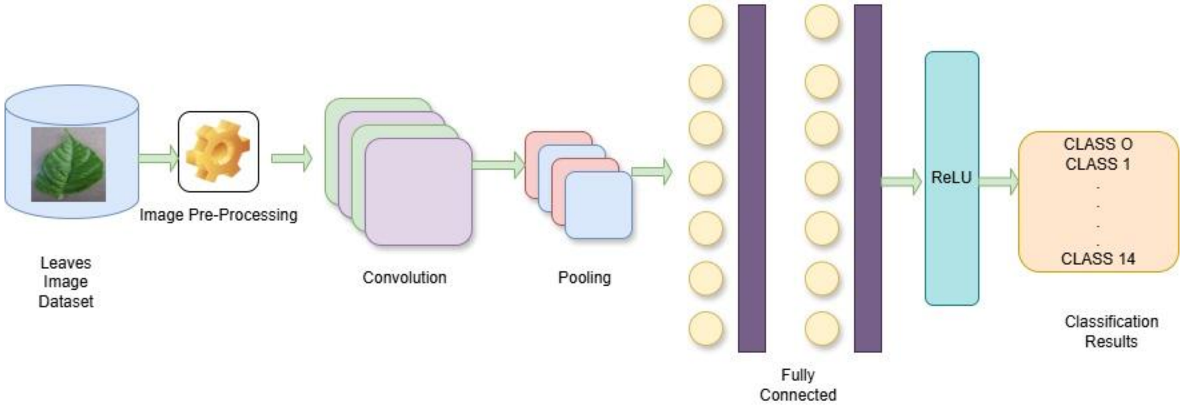


Figure 3 Fine Tuned CNN Architecture

3.3 EfficientNetB0

EfficientNet models are designed based on the principle of compound scaling, which systematically increases the size of the convolutional network to the desired dimensions. This technique utilizes a compound coefficient to uniformly scale all aspects of the network, ensuring balanced growth in width, depth, and resolution. EfficientNet employs mobile inverted bottleneck convolution blocks (MBConv) with kernel sizes of 3×3 and 5×5, which significantly reduce computation by a factor of f^2, where f is the filter size, compared to traditional convolutions. By applying the compound scaling coefficient, the network's dimensions are uniformly expanded. For our classification task, we utilized EfficientNetB0, as larger networks with increased dimensions typically result in higher accuracy.

To enhance the EfficientNetB0 model, we added a convolutional layer with 32 filters and a kernel size of 2×2, as well as a max pooling layer with a 2×2 kernel size and stride of 2, making the model more robust than the standard version. We fine-tuned parameters by replacing the final three layers to ensure the output layer matched the number of classes in the dataset. Batch normalization and L1 and L2 regularization techniques were employed to prevent overfitting. A dropout layer was also incorporated to improve the model's reliability for classification. Additionally, a dense layer with 256 neurons was added. Given the large size of the dataset, training the entire dataset posed challenges in terms of computational resources and time. Therefore, data trimming was utilized to make the model more robust and achieve high accuracy. This approach also included data augmentation to increase the diversity of the training data.

3.4 Implementation

These CNN models were trained and implemented on the Google Colab platform, leveraging the GPU, Python 3.11, TensorFlow, Pytorch, Scikitlearns, and Keras library on the Windows 11 operating system with 8GB RAM and 12th Gen Intel(R) Corei5-1235U processor. The study primarily aimed to evaluate the performance of the fine-tuned CNN model and EfficientNetB0 in classifying plant diseases. Input images were resized to align with the network's specifications and the Several output classes matched the classification layers.

Training images were processed through multiple filters at varying resolutions, with each convolutional output serving as input for the subsequent layer. As the layers progressed, increasingly complex features were extracted, effectively distinguishing leaf objects from other elements. The model was trained to classify various plant diseases into distinct categories. Each image output provided a probability score for each disease, and the model selected the disease with the highest probability as the final classification result.

Table 4 Hyper Parameters of CNN

Parameters	Values
Batch size	32
Epochs	10
Dropout	0.4
Learning Rate	0.001
Loss Function	Categorical Cross Entropy
Optimizer	Adam

This model follows the steps mentioned below Load a dataset consisting of 52 different plant disease classes. Then, employ the train_test_split function to partition the dataset into training, validation, and testing subsets. Allocate 90% of the data for training purposes, while the remaining 10% is reserved for testing and validation. Incorporating a model by convolutional layer with 32 filters, each having a kernel size of 2 × 2.

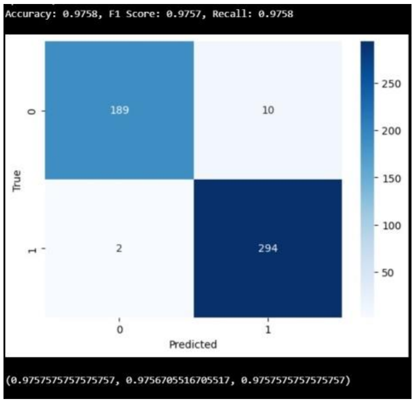
Additionally, add a max pooling layer with a kernel size of 2 × 2 and a stride of 2 to improve the model's robustness. Establish a model that takes the inputs from the base model and outputs the results from the final dense layer. Compile this model with the Adam optimizer and the categorical cross-entropy loss function. Train the model starting with a learning rate of 0.001 along with the Adam optimizer to improve training performance and control. Conduct the training process for 10 epochs and assess the model's performance using the testing dataset. Finally, compare the model's results with those of pre-trained models and make predictions on new data. It's worth noting that our model reached the desired accuracy within 10 epochs of training. Additional epochs were unnecessary. The results, which are visualized in the results section, highlight the potential effectiveness of our proposed approach.

4 RESULTS AND DISCUSSION

This part of the study focuses on evaluating the performance of our deep learning models against the dataset, considering metrics like accuracy, precision, recall, and the F1 score. We will compare our CNN models and present the outcomes of this research visually and by using some evaluation metrics such as Confusion Matrix, Accuracy, Recall, F1-score.

Through careful examination of various pre-trained CNN models and the EfficientNetB0 model using the test dataset, we determined that the EfficientNetB0 model exhibited the highest accuracy among those two models evaluated in our study. The proposed EfficientNet Model achieves the accuracy F1-score and Recall as of 0.9975, 0.9975, and 0.9975 respectively.

Compared with our tuned CNN model which achieves the accuracy, F1-score and recall as 0.9629,0.9628,0.9629respectively.By evaluating the model with the Bell Pepper plant leaves dataset, it achi eves an accuracy of 0.9757 with minimal loss which are shown in Figure 4.



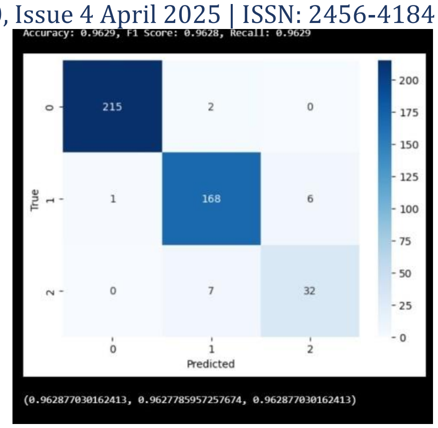
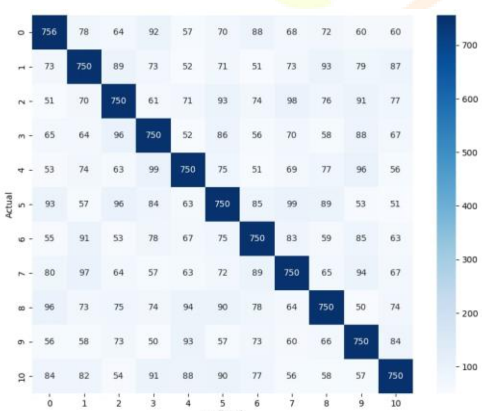


Figure 4Confusion Matrix of BellPepper CNN

Figure 5 Confusion Matrix of Potato CNN

By analyzing the performance of the model utilizingthe potato plant leaves dataset, we observed anaccuracy of 0.9628 which outperforms other pre-trained CNN models as shown in Figure 5.

Through our analysis of the model's performance using the tomato plant leaves dataset, we noted an impress ive accuracy of 0.9975, surpassing other pre-trained CNN. Figure 6 shows the confusion matrix for the tomato leaves dataset which has 10 classes.The Proposed EfficientNetB0 achieves the least loss compared to our Fine-tuned CNN model as shown in Figure 7.



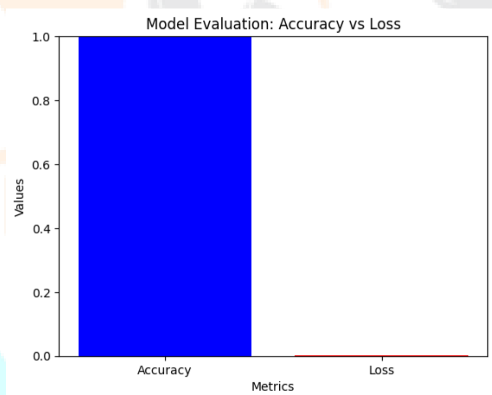


Figure 6 Performance Evaluation of EfficientNetB0

Figure 7 Confusioon Matrix of Tomato CNN

5 CONCLUSION

This study employed transfer learning and deep learning (DL) methods for classifying plant diseases. The pr oposed EfficientNetB0 model was evaluated against the Fine-tuned CNN model by fine-tuning hyper parameters, we aimed to optimize performance. By enhancing the proposed model adding the convolutional and max pooling layers. This change can played a significant role in achieving high accuracy in plant disease classification. The model does have some constraints, such as requiring a robust GPU to reduce the lengthy training duration, susceptibility to noise leading to possible misclassifications, and the necessity for real-time field testing. Future studies should aim to evaluate the models using real-time environmental images to mitigate these issues and integrate the latest developments in deep learning to boost the model's precision and efficiency.

REFERENCES

[1] Chunduri Madhurya, Emerson Ajith Jubilson "YR2S: Efficient Deep Learning Technique for Detecting and Classifying plant Leaf Diseases".

[2] Faiqua Adnan, Mazhar Jawad Awan, Amena Mahmoud, Haitham Nobanne, Awais Yasin and Azlan Mohd Zain "EfficientnetB3-Adaptive Augmented Deep Learning(AADL) for Multi-Class Plant Disease Classification".

[3] Diana Susan Joseph, Pranav M Pawar, and Kaustubh Chakradeo "Real-Time Plant Disease Dataset Development Detection of Plant Disease using Deep Learning".

[4] Emmanuel Moupojou, Florent Retraint, Hyppolite Tapamo, Marcellin Nkenlifack, Cheikh Kacfah, and Appolinaire Tagne "Segment Anything Model and Fully Convolutional Data Description for Plant Multi-Disease Detection on Field Images".

[5] Munir Majdalawieh, Shafaq Khan and MD.T.Islam " Using Deep Leaning Model to Identify Iron Chlorosis in Plants".

[6] Yang Liu, Guoqin Gao, Zhenhui Zhang " Crop Disease Recognition based on Modified Light-Weight CNN with Attention Mechanism".2022,10.1109/ACCESS.2022.3216285.

[7] Minah Jung, Jong Soeb song, Ah-Young Shin, Boemjo Choi, Sangjin Go, Suk-Yoon Kwon, Juhan Park, Sung Goo Park and Yong-min-kim" Construction of deep learning-based disease detection model in Plants",2023

[8] Wubetu Barud Demilie "Plant disease Detection and Classification Techniques: A Comparative Study of Performance", 2024.

[9] B.R. Pushpa, V.V.Aishwarya "Tomato leaf Disease Detection and Classification using CNN", 2022

[10] K.P Asha Rani, S.Gowri Shankar "Pathogen-based Classification of Plant Diseases: A Deep Transfer Learning Approach for Intelligent Support Systems.", 2023.

[11] Aanis Ahmad, Aly El Gamal and Dharmendra Saraswat " Towards Generalization of Deep learning-based Plant Disease Identification Under Controlled and Field Conditions.",2022.

[12] Jyotismita Chaki and DibyaJyoti Ghose "Deep learning in Leaf Disease Detection(2014-2024): A Visualization based Bibliometric Analysis",2024.

[13] Syed Mujtaba Hassan Rizvi, Asma Nazeer, Shafiq Ur Rehman, Sheeraz Akram and Volker Gruhn "Revolutionizing Agriculture: Machine and Deep Learning Solutions for Enhanced Crop Quality and Weed Control." 2023

[14] Sethil Kumar Swami Durai, Mary Divya Shamili " Smart Farming using Machine Learning and Deep Learnig Techniques." 2022.

[15] Amer Tabbakh and Sowbhagya Sankar Barpanda" A Deep Features Extraction Model Based on the Transfer Learning Model and Vision Transformer TLMViT for Plant disease Classification" 2023.

[16] Jun Liu and Xuewei Wang " Plant diseases and pests detection based on Deep Learning: A Review" 2021.

[17] Sanasam Premananda Singh, KeishamPrithamdas, KharibamJilenkumari Devi, Salam Devayani Devi " Custom Convolutional Neural Network for Detection and Classification of Rice Plant Diseases." 2023.

[18] Manowarul Islam, Abdul Ahad Adil, Alamin Talukder, Khabir Uddin Ahamed, Ashraf Uddin, Kamran Hasan, Selina Sharmin, Mahbubur Rahman, Sumon Kumar Debnath "DeepCrop: Deep Learning based Crop Disease Prediction with Web Application." 2023.