Grenze International Journal of Engineering and Technology, June Issue

Classification of Corn Leaf Disease using Resnet18, Alexnet and VGG16

Seema Vishwakarma[1] and Dr. Sachin Bhoite[2]
[1]Assistant Professor, Vidyalankar School of Information Technology Mumbai, Maharashtra
Email: seema.vishwakarma@mitwpu.edu.in
[2]Associate Professor Dr. Vishwanath Karad, MIT World Peace University, Pune, Maharashtra y
Email: sachintenjuly@gmail.com

***Abstract*—Corn leaf disease detection models aim to accurately identify and diagnose diseases in crops, enabling early intervention to prevent spread and minimize losses. By providing timely and precise information, these models help farmers reduce the use of chemical inputs, increase productivity, and promote sustainable agricultural practices. The ultimate goal is to enhance crop health, yield, and profitability while minimizing environmental impact. To achieve the specified goals in corn leaves, a Corn Leafe Disease Detection Model has been developed utilizing three distinct algorithms: ResNet18, VGG16, and AlexNet. Algorithms were carefully chosen due to their effective performance in similar tasks. The model was constructed using a pre-trained dataset obtained from Kaggle website consisting of images of corn leaves with disease as well as healthy leaves, ensuring a robust foundation for accurate disease detection in corn crops. The Kaggle website provided a pre-trained dataset with pictures of both healthy and infected corn leaves, which was used to build the model. This dataset ensured a robust foundation for accurate disease detection in corn crops. The dataset underwent training, validation, and testing using deep learning algorithms. Results were analyzed based on accuracy and loss measures to assess the performance of the model. A number of scholars have used deep learning and machine learning approaches to create models for identifying crop conditions. However, the comparison conducted in this paper primarily focuses on the results of validation testing to make decisions regarding the selection of algorithms for detection purposes. Among the three algorithms evaluated, VGG16 has consistently demonstrated high performance determined by precision, loss, confusion matrix, precision, recall and F1 score.**

I. INTRODUCTION

As per ICAR, maize, also known as corn, holds the third position among India's most important cereal crops, following rice and wheat. Globally, maize is highly esteemed for its versatile uses in food, feed, fodder, and various industrial applications. Since 1961, India has consistently been among the top 10 maize producers worldwide and presently holds the 6th position, with an annual production of 31.65 million metric tons.[24]. The research aims to support agricultural networks by employing deep learning algorithms to achieve optimal results. These results are crucial for the timely identification of diseases, facilitating necessary interventions. Furthermore, the ability to compare outcomes from three different algorithms aids farmers in making informed decisions. The Kaggle website provided the dataset utilized in this investigation which comprises data from the PlantVillage and PlantDoc datasets, which are widely recognized in the field. It has a size of 169 MB, indicating

Grenze ID: 01.GIJET.10.2.575
© Grenze Scientific Society, 2024

a substantial volume of information. The dataset has received notable attention on Kaggle, as evidenced by its significant upvotes. Its popularity underscores its relevance and reliability for research purposes. This dataset forms a valuable resource for investigating plant-related phenomena and developing predictive models.. The dataset was utilized to develop a predictive model within a Kaggle notebook. The dataset was employed to construct a predictive model using a notebook on Kaggle. Three distinct algorithms were utilized to forecast outcomes. This approach enables farmers to compare the results generated by each algorithm. By comparing these results, farmers can gain insights into the severity of diseases affecting their crops. This information empowers them to take timely and appropriate measures to mitigate crop losses. ResNet18, which is one of the type of deep learning algorithm based on neural network, uses residual connections to tackle the vanishing gradient problem. It comprises 18 layers and is an advanced model. In contrast, AlexNet, which consists of eight layers, was a pioneering model that introduced techniques such as ReLU activation and dropout regularization, resulting in enhanced image recognition accuracy. VGG16, with its 16 layers and small filter sizes, is renowned for its simplicity and depth, enabling the generation of rich feature representations.This paper presents a comparative study of three algorithms, each undergoing training, validation testing, and testing with a dataset split of 70:20:10. Results are compared using accuracy and loss metrics to analyze performance.

II. LITERATURE REVIEW

Ahmed et al.2021 in his paper introduces a mobile-based system utilizing Deep Learning to detect plant leaf diseases, with the intention of helping farmers with little resources identify crop diseases. The program identified 38 ailment classes across 14 varieties of crops with an astounding 94% overall classification accuracy, showcasing the system's effectiveness in plant disease detection[23].Jadhav et al. (2020) employed a five-fold cross-validation strategy to identify soybean leaf diseases using AlexNet and GoogleNet. They achieved accuracies of 98.75% and 96.25% respectively with the AlexNet and GoogleNet models [15].Hatem et al., 2023; has created a CNN model on maize crop and made use of transfer learning.The effectiveness of transfer learning models, including GoogleNet, AlexNet, ResNet50, and VGG16, is evaluated using six metrics: accuracy, precision, specificity, recall, F-score, and time. MATLAB programming software is utilized for designing and training these transfer learning models. Ma et al., 2022 has shown that while deep CNNs are powerful, lightweight networks designed for mobile terminals have shown comparable performance, which is particularly useful when dealing with small-scale datasets or when computational resources are limited [17]. David et al., 2021 in their work of Hybrid architectures combining CNNs with other neural network models, such as RNNs, have also been explored to improve early detection capabilities [19]. In summary, CNN-based methods for leaf disease identification are highly effective, with the potential to significantly contribute to agricultural production and food safety. Tugrul et al., 2022 in his research indicates that these methods can achieve high accuracy, and the use of transfer learning and lightweight networks can further optimize performance for practical applications[21]. As per Rani et al., 2021, advancements in CNN architectures and their successful application across different crops underscore the robustness of this approach in plant disease detection [20].The comparative effectiveness of ResNet, AlexNet, and VGG in identifying corn leaf diseases has been explored in several studies. ResNet and AlexNet have been shown to achieve high accuracy rates in the automatic identification of corn leaf diseases. Wu, 2021 in his work of a two-channel Convolutional Neural Network (CNN) based on VGG and ResNet achieved a validation accuracy of 98.33% for corn leaf disease identification, outperforming the VGG model alone[6]. Micheni et al., 2023 proposed that AlexNet achieved a high average accuracy of 98.3% in a study conducted in Kenya, followed by ResNet-50 with 96.6% accuracy [7]. Zeng et al., 2022 introduced a modified ResNet-50 model, SKPSNet-50, which showed an average identification accuracy of 92.9%, outperforming several other deep learning models [8]. Setiawan et al., 2021 found that Alexnet is superior to Squeezenet in a comparative study, with an accuracy of 97.69% [9]. M. Kumar et al;2020; worked on the identification of illnesses through appropriate segmentation of the afflicted area according to the kind of plant family.Convolution Neural Network (CNN), which was utilized to identify the coffee leaf disease in the article, has been utilized to detect coffee leaf diseases [17]. Convolutional Neural Networks have demonstrated their efficacy and accuracy in the phases of pattern recognition and picture categorization. The main goal of this neural network approach is the very precise identification of disease that affects coffee plant’s leaves. When compared to other ways, this procedure takes longer.

III. DATASET

Building a robust model for classifying corn leaves requires a substantial amount of high-quality data. With a dataset sourced from Kaggle, compiled using PlantVillage and PlantDOc, consisting of 4188 photos of corn leaves, the groundwork is set. This dataset encompasses various angles and sizes, providing a diverse representation. Among the most common diseases affecting corn plants—common rust, gray leaf spot, and blight—there are 1306 images of common rust, 574 of gray leaf spot, 1146 of blight, and 1162 healthy images. The approach entails a phased training process. Initially, 70% of the dataset, comprising all disease categories and healthy images, will be utilized for model training. This phase is important as it sets the base for the model's understanding of different leaf conditions. Following training, 20% of the dataset will be reserved for validation testing. This phase serves multiple purposes. It evaluates the model's performance, ensuring it generalizes well to unseen data, and helps identify overfitting issues. By simulating real-world testing conditions, validation testing provides insights crucial for refining the model. Finally, 10% of the dataset will be reserved for testing the model. This independent dataset aids as a litmus test for the model's efficiency. It allows for a comprehensive assessment of how well the model performs in real-world scenarios. Throughout this process, careful attention will be paid to preprocessing the images to ensure consistency and optimize model performance. By validating and testing the model with separate datasets, it ensures the model's robustness and effectiveness in practical applications.

IV. PRE-PROCESSING

The model is created using PyTorch, an open-source machine learning package designed in Kaggle Notebook by Facebook's AI Research group, is used to generate the CNN model. For data science, machine learning, and data analysis tasks, Kaggle offers an interactive coding environment called a notebook. Writing Python deep learning models and testing them is comparatively simple for developers.

A. System configuration

In the preprocessing stage, every image was set to have a uniform 224 x 224 pixel size, using a built-in function in PyTorch. The system environment consists of a disk space of 4.5GB, RAM with a maximum capacity of 29GB, and a GPU with a P100 configuration. CUDA is employed in Kaggle Notebooks to leverage GPU acceleration for computations. GPUs excel in parallel processing tasks, making them ideal for training deep learning models that require intensive matrix calculations and optimization algorithms.

B. Augmentation

Before the model undergoes training, the images in the deep CNNs classifier dataset are preprocessed to enhance feature extraction and consistency. Changing the image format and size is one of the most crucial responsibilities. One of the most crucial tasks is to format and normalize the image size.. Since there are differences in the quantity of each kind of leaf picture, an uneven distribution of image types could cause the model to deviate after training under various conditions. In addition to increasing the training set, balancing the variety of image types, and improving image realism, data augmentation can also prevent overfitting. The photos of the crop vary in quality and detail as the leaves depending upon many factors like overlapping of leaves, amount of light, camera quality. As a result, augmentation of data is performed to decrease overfitting and make the model to ge trained in a more realistic manner. Applying transformations.Every image in the datasets has been scaled to 224 × 224 pixels using the *transforms.Compose()* tool. After that, the photos are transformed into PyTorch and their mean and standard deviation are used to normalize them. The size of the images was adjusted to match the neural networks' input size. After normalizing the picture data, we train and assess the models. To scale the data to a range that is suitable for the network, each image is normalized. Image normalization produces contrast stretching, which improves the dataset's low contrast photos as well

C. Alexnet

Krizhevsky et al., 2017; created Alexnet consisting of two fully connected layers with five convolutional layers [24]. After a few convolutional layers, there are max-pooling layers included. AlexNet employed overlapping pooling in place of conventional layers of pooling with non-overlapping regions, which helps to improve translation invariance and lessen overfitting. In the layers that were completely linked, dropout was employed to avoid overfitting. During training, dropout randomly removes units from the neural network along with their connections, assisting in preventing overfitting. In the figure 1 given below five convolutional layers make up the design, with Max-Pooling layers included in the first, second, and fifth levels to ensure appropriate feature extraction. The Maximum pooling layers are overlapping, with a filter size of 3x3 with strides of 2. Comparing this to non-overlapped Maximum pooling layers Two fully-connected, dropout-prone layers follow them, and a SoftMax layer for predictions comes last.

D. Resnet18

The 18 layers of ResNet-18 give rise to its name. Most of these layers include of identity blocks, fully linked layers, pooling layers, and convolutional layers..The usage of residual blocks is the main innovation of ResNet. Deeper layers in a conventional deep network may experience performance decrease or disappearing gradients. Residual blocks allow the network to acquire residual functions according to the layer inputs, as opposed to learning unreferenced functions.

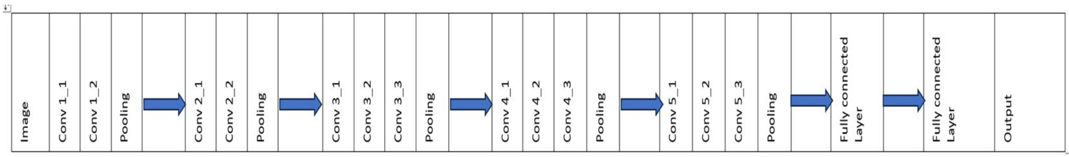

Figure 1. Alexnet Architecture

E. VGG-16

Simonyan and Zisserman[25] in their worked on VGG-16 architecture and created it. The Visual Geometry Group at Oxford University introduced it[25]. The simplicity and consistent architecture of VGG-16 make it simple to comprehend and use.Usually, the VGG-16 design has 16 layers overall—13 convolutional layers and 3 fully linked layers. These layers are arranged into blocks, and each block has a layer with the maximum pooling for down sampling after a number of convolutional layers. The VGG-16 architecture, designed for image classification tasks by the Visual Geometry Group at the University of Oxford, comprises a series of convolutional and pooling layers. The input layer, with dimensions of 224x224 pixels and three color channels, initiates the network. The input is processed via convolutional layers with 64 3x3 filters, which are then followed by same-padding to preserve spatial dimensions. The feature maps are downsampled via a max-pooling layer with a 2x2 window and stride 2 after each convolutional layer stack. Repeating this method with higher filter counts results in blocks that have convolutional layers with 128 and 256 filters in them, but with the same padding and 3x3 filter size. Three sets of 512-filter convolutional layers each highlight the depth of the architecture, which is then further enhanced by max-pooling layers. Subsequently, two more convolutional layers with 512 filters each are used, and the result is then flattened into a vector with a size of 25088. Finally, three fully linked layers are used for classification; each layer has 4096 neurons and a softmax activation., are employed for classification, with the final layer outputting probabilities for 1000 classes in the Imagenet Large Scare Visual Recognition challenge.

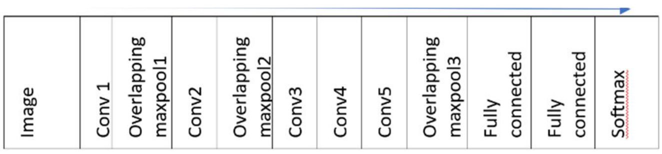

Figure 2. VGG-16 Architecture

V. Methodology

In the data preprocessing phase, images were prepared for model input by resizing them to 224x224 pixels, converting them to tensors, and normalizing them using ImageNet's mean and standard deviation values. This standardization process ensures that the model receives inputs in a consistent format, enhancing its ability to learn effectively. During the model selection phase, three popular algorithms—VGG16, ResNet-18, and AlexNet—were benchmarked. Among these, VGG16 exhibited superior performance, characterized by consistent accuracy and stable reduction in loss across the validation dataset to tailor the model to our specific problem statement, the classifier layer of VGG16 was modified to output four classes. Following this modification, the entire model underwent fine-tuning on our dataset, optimizing its parameters to better suit our classification task. The dataset was divided into three subsets for training, validation, and training sets. During the training phase, the cross-entropy loss function was used. The training dataset's mean and standard deviation were also computed during the training phase. In order to guarantee that every feature contributes equally to the learning process and to keep features with bigger scales from predominating in the training, these statistics were utilized to normalize the input data. Using the conventional formulas, the mean and standard deviation were determined. In order to ensure that the model generalizes effectively to new data, a validation step was also incorporated into the training loop to track the model's performance and reduce overfitting. Ultimately, an unidentified test dataset was used to assess the model's performance. The generalization capacity and overall performance of the model were evaluated using both accuracy and loss criteria. The proportion of accurately categorized photos to all images is known as accuracy. while standard deviation provides a measure of the spread of the data, giving insight into the variability of the model's predictions.

VI. RESULTS

The validation accuracy, loss, and confusion matrix were used to predict the results. Detailed comparisons of accuracy are presented in Figure 3, and comparisons of loss are shown in Figure 4. While all three algorithms performed well individually, VGG16 consistently outperformed the others. AlexNet exhibited significant fluctuations, particularly noticeable around epoch 7, while ResNet-18 maintained consistent but relatively lower accuracy below 0.96 across all epochs. In contrast, VGG16 consistently maintained high accuracy without major fluctuations, indicating its robust performance.In terms of loss metric, both AlexNet and ResNet-18 displayed higher levels of loss compared to VGG16. VGG16 showed lower loss values consistently, indicating its better performance in minimizing prediction errors. This suggests that VGG16 is more effective in reducing the discrepancy between predicted and actual values compared to AlexNet and ResNet-18. Comparing the performance of AlexNet, ResNet-18, and VGG16, all three models exhibited strong performance in terms of test accuracy and precision. With a test accuracy of 0.9431, VGG16 and AlexNet both came in second and third, respectively, behind ResNet-18, which had the best test accuracy of 0.9550. However, VGG16 and ResNet-18 achieved slightly lower test losses of 0.1631 and 0.1859 respectively, compared to AlexNet's test loss of 0.1599. Regarding precision, ResNet-18 had the highest value of 0.9454, followed closely by AlexNet at 0.9399 and VGG16 at 0.9417. AlexNet scored the highest in recall (0.9486), followed by VGG16 (0.9468) and ResNet-18 (0.9363). On the other hand, when recall and precision are balanced, the F1 score, VGG16 performed slightly better with a score of 0.9438. followed closely by AlexNet at 0.9431 and ResNet-18 at 0.9404. Overall, all three models demonstrated strong performance, but VGG16 exhibited a slightly more balanced performance across various metrics.

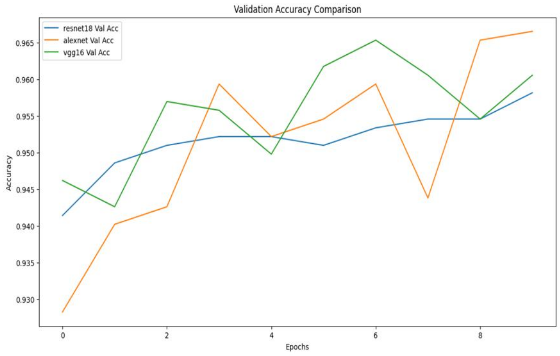

Figure 3. displaying the results of comparison of validation accuracy of resnet18,vgg16 and alexnet in each epoch

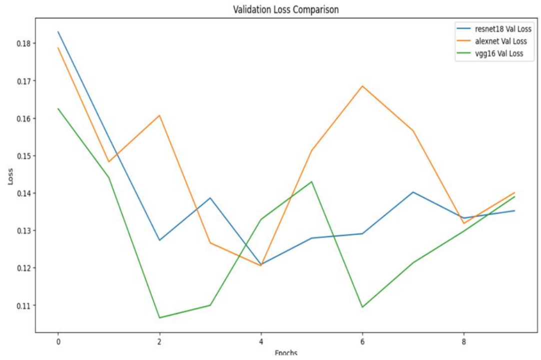

Figure 4. displaying the results of comparison of validation loss of resnet18,vgg16 and alexnet in each epoch

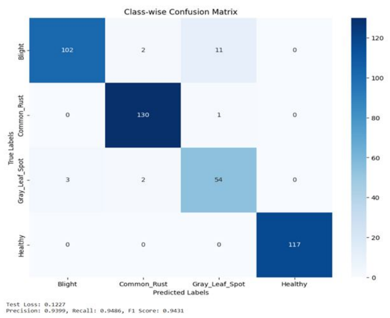

Figure 5. Confusion matrix for alexnet model

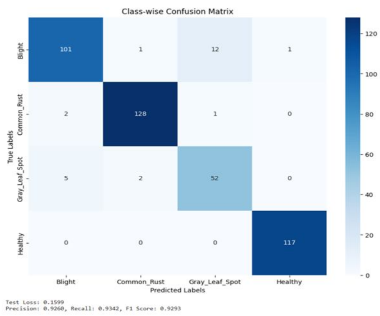

Figure 6. Confusion matrix for resnet18 model

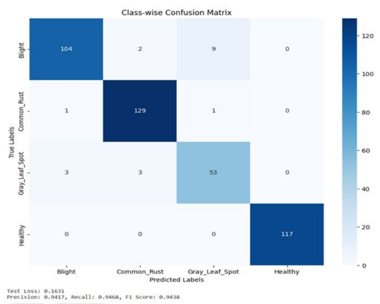

Figure 7. Confusion matrix for Vgg16 model

VII. Conclusions

The corn leaf disease detection model successfully ran and accurately predicted results using three different algorithms. These algorithms underwent testing on a large dataset sourced from Kaggle, evaluating various metrics including accuracy, loss, and the confusion matrix. Each algorithm exhibited strengths in different aspects, but upon overall comparison, VGG16 emerged as the most effective for detecting corn leaf diseases. While all three algorithms demonstrated competence, VGG16 consistently outperformed others across multiple evaluation criteria. Its superior performance suggests that VGG16 is more adept at accurately identifying and classifying corn leaf diseases in comparison of the other algorithms. This finding emphasizes the importance of choosing the right algorithm for the task at hand. VGG16's success in this context highlights its potential for application in agricultural disease detection systems, indicating its suitability for addressing real-world challenges in crop health monitoring and management.

References

[1] Samit Hatem, A., Altememe, M.S., & Fadhel, M.A. (2023). Identifying corn leaves diseases by extensive use of transfer learning: a comparative study. Indonesian Journal of Electrical Engineering and Computer Science.

[2] Srivastava, P., Singh, S.P., Vishnoi, A., & Sapra, V. (2022). Classification of Corn Leaf Diseases using Various Pre-trained Deep Learning Networks and Performance Comparison, 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM), 1-5.

[3] Hu, R., Zhang, S., Wang, P., Xu, G., Wang, D., & Qian, Y. (2020). The identification of corn leaf diseases based on transfer learning and data augmentation, Proceedings of the 3rd International Conference on Computer Science and Software Engineering.

[4] Abdul Waheed, Muskan Goyal, Deepak Gupta, Ashish Khanna, Aboul Ella Hassanien, Hari Mohan Pandey,An optimized dense convolutional neural network model for disease recognition and classification in corn leaf,Computers and Electronics in Agriculture,Volume 175, 2020, 105456, ISSN 0168-1699, https://doi.org/10.1016/j.compag.2020.105456.

[5] Yousef Methkal Abd Algani, Orlando Juan Marquez Caro, Liz Maribel Robladillo Bravo, Chamandeep Kaur, Mohammed Saleh Al Ansari, B. Kiran Bala,Leaf disease identification and classification using optimized deep learning,Measurement: Sensors,Volume25, 2023, 100643, ISSN 2665-9174, https://doi.org/10.1016/j.measen.2022.100643.

[6] Yuhao Wu, Identification of Corn Leaf Diseases based on Convolutional Neural Network, 2021 J. Phys.: Conf. Ser. 1748 032004

[7] Micheni, M., Birithia, R.K., Mugambi, C., Too, B.K., & Kinyua, M. (2023). Identification of Maize Leaf Diseases Based On AlexNet and ResNet50 Convolutional Neural Networks. Indonesian Journal of Computer Science.

[8] Zeng, W., Li, H., Hu, G., & Liang, D. (2022). Identification of maize leaf diseases by using the SKPSNet-50 convolutional neural network model. Sustain. Comput. Informatics Syst., 35, 100695.

[9] Setiawan, W., Ghofur, A., Hastarita Rachman, F., & Rulaningtyas, R. (2021). Deep Convolutional Neural Network AlexNet and Squeezenet for Maize Leaf Diseases Image Classification. Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, 6(4). https://doi.org/10.22219/kinetik.v6i4.1335

[10] Zhang, X., Qiao, Y., Meng, F., Fan, C., & Zhang, M. (2018). Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks. IEEE Access, 6, 30370-30377.

[11] ANAK ENTUNI, C.J., & Zulcaffle, T.M. (2022). Identification of Corn Leaf Diseases Comprising of Blight, Grey Spot and Rust Using DenseNet-201. Borneo Journal of Resource Science and Technology.

[12] Bhatt, Prakruti, Sarangai, Sanat, Shivhare, Anshul, Singh, Dineshkumar, Pappula, Srinivasu, 2019. Identification of diseases in corn leaves using convolutional neural networks and boosting. In: International Conference on pattern recognition applications and methods, Feb. 2019

[13] Li, L., Zhang, S., & Wang, B. (2021). Plant disease detection and classification by deep learning—a review. IEEE Access, 9, 56683-56698.

[14] Sodha, N., Chanda, K., Rawate, S., Chaudhary, S., Suryawanshi, R., & Vazalwar, N. (2022). Plant Leaf Disease Prediction Using Deep Learning (No. 7569). EasyChair.

[15] Jadhav, S. B., Udupi, V. R., & Patil, S. B. (2021). Identification of plant diseases using convolutional neural networks. International Journal of Information Technology, 13(6), 2461-2470.

[16] Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of apple leaf diseases based on deep convolutional neural networks. Symmetry, 10(1), 11.

[17] Ma, Z., Wang, Y., Zhang, T., Wang, H., Jia, Y., Gao, R., & Su, Z. (2022). Maize leaf disease identification using deep transfer convolutional neural networks. International Journal of Agricultural and Biological Engineering, 15(5), 187-195.

[18] Parvez, S., Uddin, M. A., Islam, M. M., Bharman, P., & Talukder, M. A. (2023). Tomato leaf disease detection using convolutional neural network.

[19] David, H. E., Ramalakshmi, K., Venkatesan, R., & Hemalathad, G. (2021). Tomato leaf disease detection using hybrid CNN-RNN model.
[20] Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Ijaz, M. F., & Woźniak, M. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. Sensors, 21(14), 4749.
[21] Tugrul, B., Elfatimi, E., & Eryigit, R. (2022). Convolutional neural networks in detection of plant leaf diseases: A review. Agriculture, 12(8), 1192.
[22] Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N. PlantDoc: a dataset for visual plant disease detection. InProceedings of the 7th ACM IKDD CoDS and 25th COMAD 2020 Jan 5 (pp. 249-253).
[23] Ahmed, A. A., & Reddy, G. H. (2021). A mobile-based system for detecting plant leaf diseases using deep learning. AgriEngineering, 3(3), 478-493.
[24] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90.
[25] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[26] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
[27] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In Advances in Neural Information Processing Systems (NIPS), 2012.
[28] M. Kumar, P. Gupta, P. Madhav and Sachin, "Disease Detection in Coffee Plants Using Convolutional Neural Network," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 755-760, doi: 10.1109/ICCES48766.2020.9138000.
[29] https://iimr.icar.gov.in/?page_id=111