International Journal of
Technology and Emerging Research
ISSN: 3068-109X (online) | ISSN: 3068-1995 (print) | https://www.ioro.org/ijter

Performance Visualization of a Self-Devised HMBFNet for Disease Detection in Plants for Their Earlier Diagnosis

Shivi[1], **Shaveta Kalsi**[2]

[1]*Research Scholar*, [2]*Assistant Professor*

[1,2]*DAV Institute of Engineering and Technology*

Abstract

Plant disease is one of the primary issues that directly reduce the quality of agricultural produce. Identification and classification of plant diseases are among the primary tasks to improve the overall quality of crop production for economic development. Numerous approaches for disease detection have been demonstrated by a group of researchers using a digital image dataset of plant leaves. However, due to their intricate algorithms and incapacity to provide an effective method for clearly demarcating boundaries among the provided data classes for the purpose of making the ultimate decisions, such systems fall short of the anticipated perfection. Deep Neural networks, which are multi-layer architecture models whose layers learn to represent the data at several abstract levels and have a less complex algorithm compared to conventional statistical methods, can be suggested as a solution for this problem. In the present context of the problem, a self-designed Multi-Branch Fusion Network model for the multiclass categorization of plant diseases has been proposed in this work. Images from several plant diseases were taken from a publicly accessible dataset and used to simulate the proposed model. It has been discovered that the suggested method works well in the specified situation.

Keyword: Plant Diseases, Multiclass, Classification and Self devised Model.

1. Introduction

Indians get revenue mostly from agriculture. They are totally dependent on the production of horticulture products and different types of food grains. According to the latest recent data, which was released in 2021, 13497.30 million hectares are cultivated globally. India ranked eighth in the world behind the Russian Federation, Canada, the United States, China, Brazil and Australia with 328.73 million hectares (2.44%) of the country's total area devoted to agriculture [1]. India is second in the world after China in terms of rice and wheat production, contributing 23.71% of the total (749.19 million metric tons) produced worldwide. [1].

Furthermore, the population's rapid increase is driving up the demand for grain on a daily basis. According to the most recent research, the country is expected to produce a record 308.65 million metric tons of food grains in 2020–21, up 11.14 million metric tones from 2019–20. Furthermore, food grain production in 2020–21 is expected to exceed by 29.77 million metric tones the average output of the five years prior (from 2015–16 to 2019–20). 12,270,000 metric tones of grain are expected to be produced in 2020–21, setting a new record. It produces 9.83 million metric tones more than the average of 112.44 million metric tones produced over the preceding five years [1]

In recent years, India has made headway towards its goal, but the usage of pesticides has increased in an attempt to avert similar tragedies. The use of pesticides climbed to 61.70 thousand metric tones in 2019–20 from 59.67 thousand metric tones in 2018–19.

Plant diseases have caused large losses in agricultural productivity and quality, which has been a major problem for agriculture and food security. India is one of the biggest food producers in the world and the economy of the nation is greatly influenced by its agricultural sector. This affects not just the country's food security but also the livelihoods of the farmers. However, the yearly loss of crop output in India as a result of plant diseases is estimated to be between 15 and 20% [2]. This translates into an annual loss of about 45,000 crores ($6,000 million). Globally, plant diseases have a big influence on agriculture and food production. Up to 20% of crop losses worldwide, according to current statistics, are caused by plant diseases, which can have detrimental effects on the economy and food security [3].

With the world population projected to reach 9.7 billion by 2050, it is crucial to find solutions to mitigate the impact of plant diseases on crop yield and quality. The usage of pesticides to treat crops for various illnesses has expanded in an effort to lessen India's food shortage. Many machine learning-based methods for disease monitoring and detection have been introduced recently in an effort to lower input costs and apply tailored pesticide applications that benefit the environment. For instance, the author in [4] suggested a method based on image processing and machine learning for identifying inadequacies in pictures of rice plant leaves. Three plant leaf colour, texture and colour photos were examined. These characteristics were used with an MLP classifier to identify areas of deficiency, yielding 88.56% correct results. Similar to this, the author of [5] presented a software system for identifying diseases in rice plants. In this system, a threshold-based technique was used to segment the affected area of the rice plant leaf. A SOM neural network was used to identify the segmented images based on data obtained from the zooming technique. It was discovered that images in the frequency domain had a low classification rate as opposed to an original image. The author of [6] used machine learning techniques to categories and identifies plant leaf diseases. The colour, shape, and texture attributes were obtained after the images were segmented using the Ostu approaches suggested threshold settings. Using a support vector machine classifier, these features were used to categorize rice infections; the classification accuracy was 97.2%. However, 11.1% of patients were misclassified because the patterns of the two categories of diseases were remarkably similar. The author of [7] created a system for categorizing diseases that impact plants in a laboratory environment. The images were collected using a digital camera, and their classification was done using the BP neural network. This constructed classifier might potentially process 90% of all the pictures in the data set. The output of the system proved that it could effectively separate rice leaves from brown spots. The planned technology, however, was unable to distinguish between illnesses under the different natural illumination conditions. The author of [8] processed citrus fruit plant leaves to look for diseases. The citrus lesion spots are extracted using an optimised weighted segmentation technique and an upgraded input image. Then, geometric shapes, colour, and texture are combined to form a codebook. The best features are also selected using a hybrid feature selection method that incorporates entropy, PCA score, and a skewness-based covariance vector. The Multi-Class Support Vector Machine (M-SVM) receives the selected attributes as input for the final citrus disease categorization. In a comparable setting, the author of [9] created a deep learning technique to identify vine illnesses by utilising colorimetric spaces and vegetation indices from UAV pictures. The method for identifying illnesses in vine yards was based on convolutional neural networks (CNN) and colour information. In this comparison of CNN performances using different colour spaces, vegetation indices, and the combination of both information, CNNs with YUV colour space combined with ExGR vegetation index and CNNs with a combination of ExG, ExR and ExGR vegetation indices yield the best results with accuracy greater than 95.8%.

Waterlogging stress was identified by the author in [10] using hyperspectral photos of oilseed rape leaves. The article describes the creation of classification models for training and prediction, as well as for contrasting the images and spectra of samples under varying levels of water logging among the three datasets, using support vector machine (SVM), quadratic discriminate analysis (QDA) and k-nearest neighbour (KNN) classifiers. The QDA mode was found to perform better in terms of categorization, with identification accuracies of 100% and 94.44%, respectively. Overall, the classification outcomes for VNIR were better than those for images. These findings demonstrated the viability and use of hyper-spectral imaging technology for the identification of oilseed rape water-logging stress. Similar to this, the author of [11] estimated the chlorophyll content of soybean leaves in the field using machine learning and digital photos taken on a smartphone. It was found that the standard calibration board was not required when using the raw RGB input to estimate chlorophyll directly using the SVM model. The method for processing images using machine learning modeling and conversion relationships to measure the chlorophyll content of infield soybean leaves that has been developed is efficient, inexpensive, and can be readily applied to other large-scale aerial imaging platforms and field crops. It also does not require the use of a standard calibration board. The author of [12] developed a method for detecting and predicting four rice-related diseases, including rice blast, red blight, stripe blight, and sheath blight, using support vector machines and deep learning. The recommended method had a 96.8% average correct identification rate. The author [13] developed a method for identifying several leaf diseases in rice plants. The proposed algorithm uses the Moore-Penrose pseudo-inverse Weight-related DCNN technique to extract disease categories from the image data set. Existing systems such as Deep Convolutional Neural Network (DCNN), AlexNet, Convolutional Neural Network (CNN), and Support Vector Machine (SVM) with Deep Features have shown an average 4% improvement in the trial.

However, a number of techniques have been proposed previously for the early detection of different plant leaf diseases. Regretfully, none of these techniques can objectively quantify the impacted region, draw the edges with accuracy, or forecast the maximum size. Previous research has indicated that a viable answer to such a question is a deep learning technique. Therefore, it has also been employed in this work. Moreover, research on deep convolutional neural networks has progressed over time by adding novel techniques to achieve remarkable results. Deep learning models can be of two types: customized neural networks and transfer learning models. Models that have already been trained and customized neural networks both function to automatically extract features from sample images. But, in the framework of deep learning, transfer learning models are specifically the techniques that apply the properties that a network has learned from the same context of a problem to solve a variety of problems in the exact same field. Transfer learning models have a number of advantages when used, including reduced computation time and great usefulness on small datasets. However, they may encounter significant challenges when trying to solve various problems in the same domain where they were trained, including negative transfer and inaccurate decision boundaries between multiple classes in the target domain's dataset. This will make it inappropriate for real-time uses, like automatically plant diseases detection. Therefore, it is advised to construct a self-designed HMBFNet model with learning that is started from scratch in order to achieve optimal performance in the given context of the problem. The core of this study is the optimal hyper-parameter selection for reducing the network error rate of a deep learning model and then simulating the same on a publicly accessible dataset comprising images associated with various plant diseases.

2. Self-devised HMBFNet model

Four simultaneous objectives are targeted in the model's design: memory, speed, accuracy and limited computational capacity. These characteristics are dictated by the characteristics of the programme, which requires a diagnosis system to be faster and more memory-efficient than an application that need diseases detection due to its more interactive nature. Finally, the costs associated with false positives or false negatives influence accuracy in this case.

2.1. Architectural design

The architectural design of self-devised HMBFNet model has been presented in figure 1.

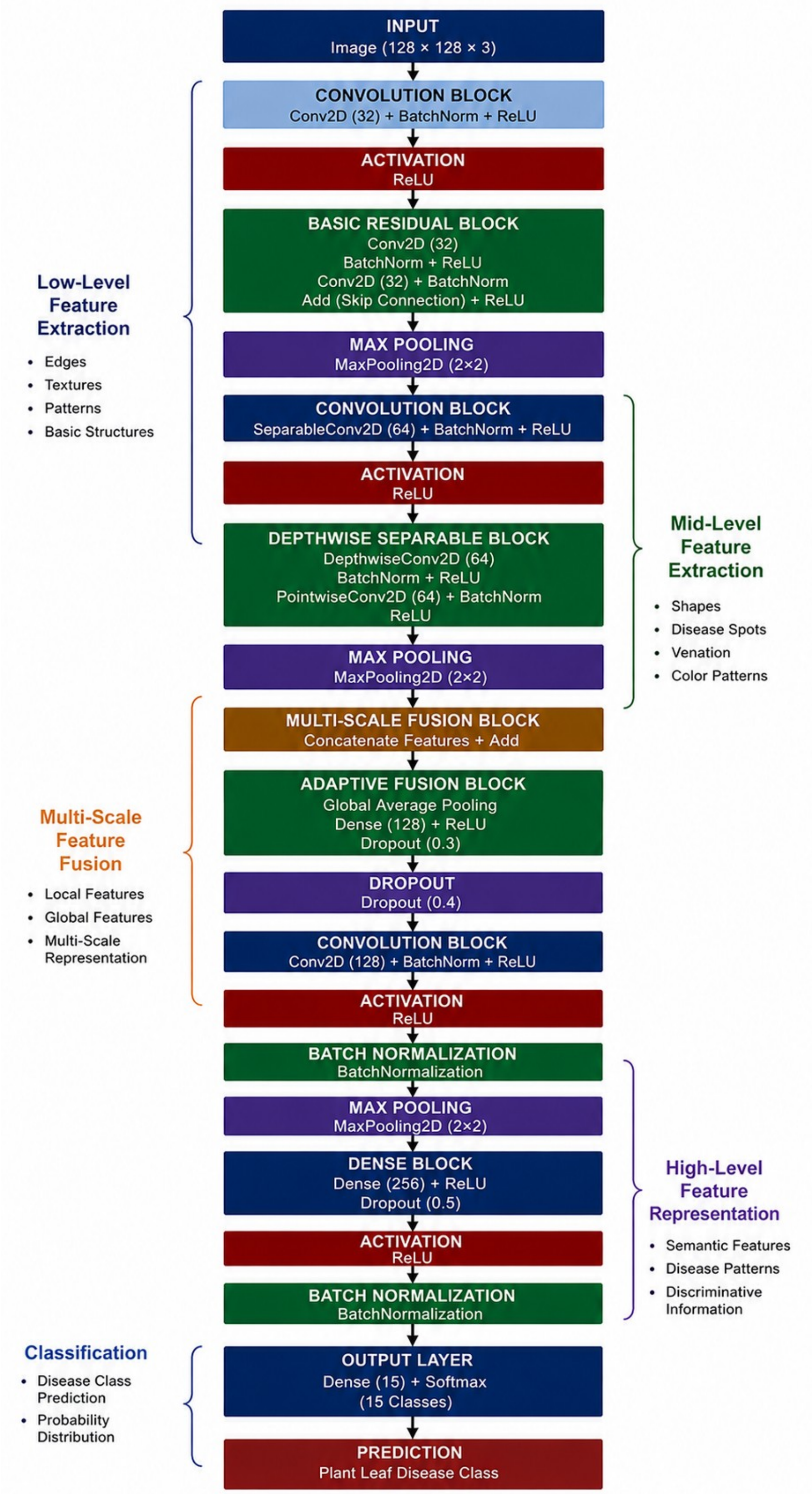


Figure 1: Architectural design of self-devised HMBFNet model

The Multi-Branch Disease Network with Adaptive Fusion is proposed to handle these issues, aiming to enhance detection accuracy, improve prediction reliability, and strengthen robustness so it works better for practical agricultural use.

The proposed Hierarchical Multi-Branch Fusion Network (HMBFNet) is a deep learning framework developed to effectively detect plant leaf disease using different level of complementary feature representations across multiple stages. At start it preprocessed RGB leaf image of size 128 × 128 × 3 as input, where preprocessing includes resizing the image, applying median filtering for noise removal, using CLAHE based contrast enhancement, normalization, and data augmentation to increase diversity in the samples, and reduce overfitting in practice. A convolutional stem block with a Conv2D layer, batch normalization and ReLU activation initially extracts low-level visual from the images including the edges textures and basic disease patterns. Further it has been enhanced by three different complementary feature extraction branches. The first branch relies on depth-wise separable convolutions and residual blocks, to efficiently capture fine texture details, the second branch mixes residual learning with dilated convolution, plus a Convolutional Block Attention Module (CBAM), for strengthen localization of diseased areas, in between getting broader context, and the third branch brings together depth-wise separable convolution, dilated convolution, and Squeeze-and-Excitation (SE) attention, learning multi-scale structural meaning of infected leaf regions.

Such discriminative features combined the three branches get merged via an Adaptive Feature Fusion and concatenates those complementary feature vectors and fed to fully connected layers, along with batch normalization and dropout. This fusion strategy helps the model put together local texture clues, global contextual cues and multi scale representations. After that, a sequence of dense layers does high level feature abstraction, before the Softmax layer finally sorts the input image into one of the fixed disease classes.

2.2. Simulation parameters

The self-devised models is trained using the following simulation parameters

Table 1: Self devised model's simulation parameters details

	Simulation Parameters		Details
1.	Epoch	---	30
2.	Batch size	---	32
3.	Learning rate	---	0.001
4.	Optimizer	---	Adam
5.	Loss	---	Sparse Categorical_crossentropy
6.	Time per epoch	---	320sec
7.	Total parameters	---	1,801,349
8.	Total learning parameters	---	1,794,629
9.	Total non- learning parameters	---	6,720
10.	Memory per total parameters	---	11.10 MB
11.	Memory per learning parameters	---	11.09 MB
12.	Memory per total non- learning parameters	---	12.75KB
13.	Train/test split	---	pre-split dataset

Significant modifications to the deep filters using convolution were not desired, which is why the modest learning rate was selected. It became observed that training with losses that erupted to very large levels and did not recover caused the procedure of learning to collapse. The batch size was selected based on the maximum value that could support positive learning of the model. This is because it is recognized that, as previously mentioned, bigger batch sizes offer a more precise calculation of the gradient for the amount of weight update. By using endpoints for each epoch during model training, the total number of epochs was ascertained. The previous epoch's endpoint was used when over fitting was identified. A pre split training and validation set was originally created in the dataset. Twenty-five percent was used for testing, and seventy-five percent was used for training the model, the same has been mentioned in the training/test splits. There were

1,794,629 parameters in total has used in training the model. This demonstrates that the computing time with the depth of the model and the number of parameters has been varied The computing cost and memory required while training the model with this reasonable count of parameters are very low, about 12MB, which shows the model proficiency from the point of view of its architectural design.

2.3. Performance testing parameters of model

The model's accuracy and loss testing parameters are essential to checking the effectiveness of the model in the given context of the problem. For image classification of plant diseases, the following performance testing parameters for the model have been selected.

Table 3: Performance testing parameters of model

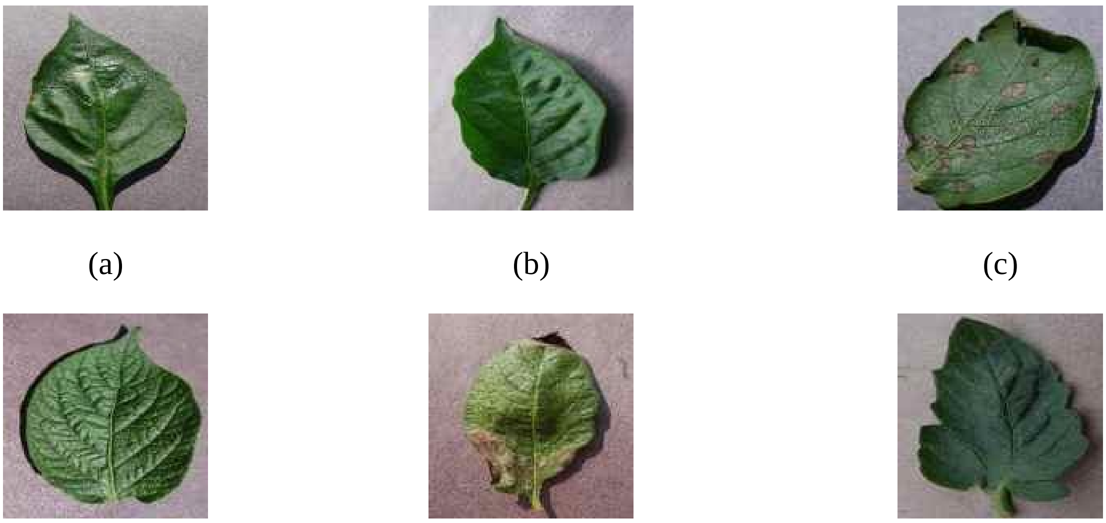
	Parameters		Value
1.	Input Shape	---	128*128 pixels
2.	Model	---	Hierarchical Multi-Branch Fusion Network (HMBFNet)
3.	Layers	---	Residual Block, Depthwise Separable Convolution, Dilated Convolution, SE Attention, CBAM
4.	Activation Function for Dense Layer	---	ReLU (Hidden Layers), Softmax (Output Layer)
5.	Loss Function	---	Categorical_crossentropy
6.	Metrics	---	Accuracy

3. Training dataset

For the simulation of the self-devised HMBFNET model, the plant leaf dataset has been used. The details of the dataset and its preparation before testing using a self-devised HMBFNET model are given below.

3.1. Dataset description

The dataset used in this work is composed of moderate resolution plant leaves diseases. The dataset consists of 22787 training images with an expansion of 256* 256 pixels. Figure 2, shows sample image in the dataset.



(a) (b) (c)

(d) (e) (f)

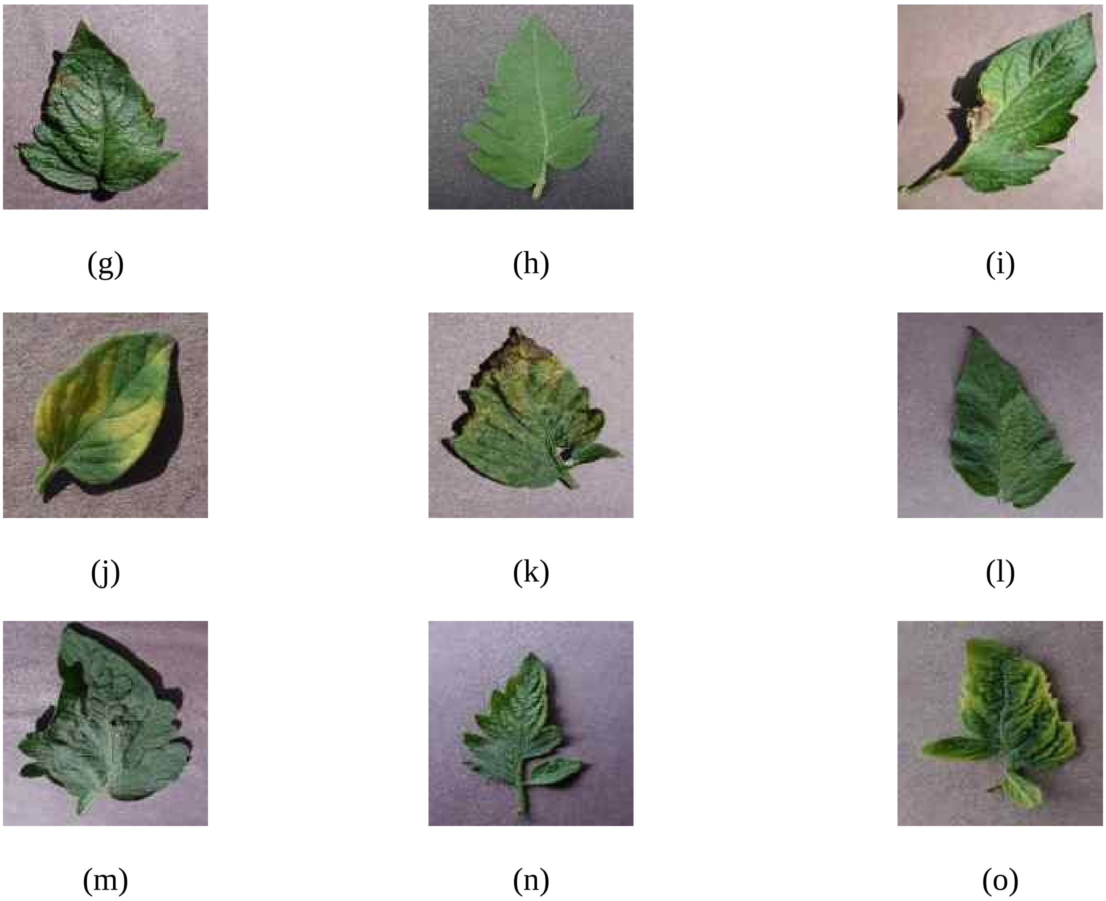


Figure 2: Sample images for rice leaf diseases (a) pepper,_bell___bacterial_spot (b) pepper,_bell___healthy (c) potato___early_bligh (d) potato___healthy(e) potato___late_blight (f) tomato___bacterial_spot (g) tomato___early_blight (h) tomato___healthy (i) tomato___late_blight (j) tomato___leaf_mold (k) tomato___septoria_leaf_spot (l) tomato___spider_mites two-spotted_spider_mite (m) tomato___target_spot (n) tomato___tomato_mosaic_virus (o) tomato___tomato_yellow_leaf_curl_virus [21]

Table 4: Dataset description

Sample No	Class Name	Total training samples	Total testing samples
0	pepper,_bell___bacterial_spot	797	200
1	pepper,_bell___healthy	1183	295
2	potato___early_bligh	800	200
3	potato___healthy	121	200
4	potato___late_blight	800	31
5	tomato___bacterial_spot	1702	425
6	tomato___early_blight	800	200
7	tomato___healthy	1272	382
8	tomato___late_blight	1527	191
9	tomato___leaf_mold	761	354
10	tomato___septoria_leaf_spot	1417	335
11	tomato___spider_mites two-spotted_spider_mite	1341	281
12	tomato___target_spot	1123	1071
13	tomato___tomato_mosaic_virus	299	72
14	tomato___tomato_yellow_leaf_curl_virus	4286	318

3.2. Pre-processing

3.2.1. Image resizing

The PlantVillage dataset that was obtained from Kaggle includes high-quality RGB images that include healthy and diseased plant leaf instances of various distinct crop species along with disease categories. In order to keep symmetry in the images, image resizing was done first as part of the preprocessing process. This can result in uniform image dimensions, hence minimizing the computational cost and memory use also. In the experimentation, the image size was resized to 128 × 128 × 3 pixels. This preprocessing step also helps keep disease signs readable after scaling like leaf spots, lesions, discoloration, mildew, rust and even edge deformation. Also, when every image has the same shape, feature extraction becomes consistent across all the branches of the fusion network that we proposed, rather than letting differences from camera resolution introduce weird variation. Hence, it overall removed size-related inconsistencies and made training stable and sped up convergence, therefore improving classification outcomes.

3.2.2. Noise Removal and Contrast Enhancement

Although the PlantVillage dataset is compose of fine quality images under controlled environmental conditions but it is can have small sensor noise illumination differences and intensity oscillations that reduce the clarity of disease symptoms. Therefore, enhancement step was used before applying the model including the noise removal and contrast enhancement. In this, a median filtering method was chosen. Its goal is to reduce impulse noise while keeping meaningful structural information, like lesion boundaries, leaf veins, and infected regions. In comparison with many ordinary smoothing filters, median filtering tends to remove stray pixel disturbances rather than “smearing” the patterns. After that, Contrast Limited Adaptive Histogram Equalization (CLAHE) was applied to increase local image contrast and highlight fine symptom differences. CLAHE improves brightness variation within limited areas, not just the whole image at once, so small lesions, fungal spots, chlorosis, necrotic tissue, and general discoloration become more obvious. It also helps reduce the effects of uneven lighting and shadow artifacts. Because of this, the deep learning model can retrieve more useful texture and color cues. Overall, the combined preprocessing strategy improves how plant leaf images look while still keeping diagnostic, clinically relevant characteristics for disease classification.

3.2.3. Image augmentation

Machine learning models need usually a varied training data to get good generalization hence data size is important. PlantVillage dataset has over thousands of labeled leaf images spanning 15 crop–disease classes, but to overcome the overfitting, image augmentation was used while training. Here random horizontal flipping and vertical flipping were used, along with rotation, zooming, width and height shifting. These augmentation moves end up producing many realistic variants of each training image, but they still keep the same original disease label. As a result, the proposed model should learn to identify diseased leaves even when the capture conditions change like different viewpoints, orientations, lighting situations, and camera distances. Also, augmentation helps with class balance, because it effectively grows the training set size for less represented disease categories without needing new data to be collected.

Table 5: Details of dataset augmentation technique

	Technique		Value
1.	Rotation_range	---	15
2.	Rescale	---	1./255
3.	Shear_range	---	0.2
4.	Zoom_range	---	0.2

5.	Horizontal_flip	---	True
6.	Fill_mode	---	Nearest
7.	Width_shift_range	---	0.1
8.	Height_shift_range	---	0.1

4. Experimental Setup

To obtain the results of the simulation using the given dataset, an experiment was carried out using a T4 GPU, which is virtually available on the Google Colab platform. The GPU is available with a limited use of 6 hours a day. Along with this, the experiment has used hardware with memory sizes of 14GB and 64GB, RAM and ROM, respectively. The model has been assessed using its accuracy and loss. The details of the virtual hardware used for this experimentation are given in Table 6.

Table 6: Virtual hardware details for experimentation

	Hardware		Details
	Platform	---	Google Colab
	GPU	---	T4
	Programming version	---	Python 3
	RAM	---	14 GB
	ROM	---	64GB

5. Results and analysis

This section has gives detailed analysis of the results for the experiment performed to detect diseases belonging to plant leaves.

5.1. Results of accuracy

5.1.1. Accuracy graph of HMBFNet

Similar to standard CNN assess, for the HMBFNet models' effectiveness, the train/validation sample was presplit. The curve representing the training and validation accuracy of the HMBFNet when simulated in the given dataset of plant leaves has been presented in figure 3.

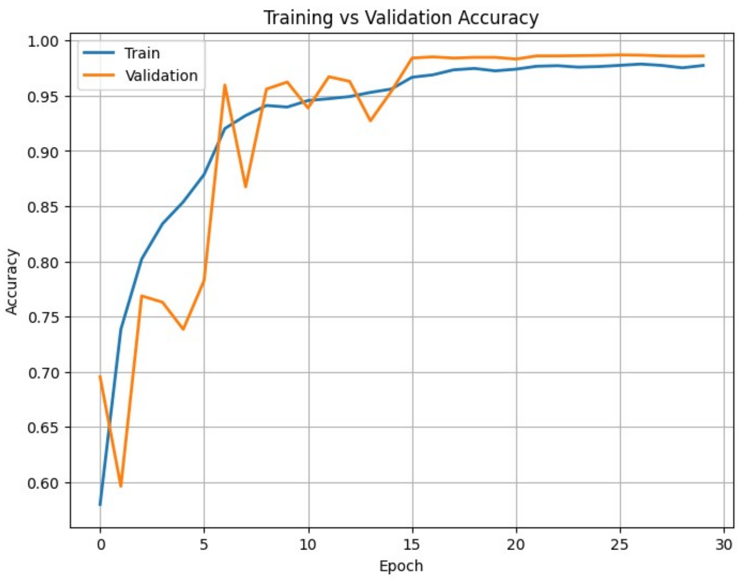


Figure 3: Accuracy curve analysis

Figure 3 shows the training and validation accuracy of the proposed Hierarchical Multi-Branch Fusion Network (HMBFNet) across 30 training epochs. In the early stage (epochs 1–5), both training and validation accuracies improved. After about the 10th epoch, the two lines start to level out, which suggests the optimization is settling into a consistent convergence. At that point, training accuracy slowly goes up from roughly 58% to 97.7%, while validation accuracy increases from around 69.5% to 98.6% by the end. The fact that the two curves are close together, implies the method generalizes well to unseen data , not showing any overfitting. Also, in the later epochs, validation accuracy stays a bit higher than training accuracy.

5.1.2. Loss graph of HMBFNetmodel

For the HMBFNetmodel, the loss curve while simulating it on the given dataset has been shown in figure 4

Figure 4: Loss curve

Figure 4 shows how the training and validation loss behave for the proposed Hierarchical Multi-Branch Fusion Network (HMBFNet) for 30 epochs. Early on, both losses stay rather high, mostly because the weights are still random. As optimization improved, the training loss keeps dropping, roughly from 1.38 down to 0.07, suggesting that the network actually learns useful feature representations from the input images. Around the 15th epoch or so, both curves starts reducing. The validation loss levels reaches to 0.04, and the training loss stays close to 0.07. Overall the validation loss being slightly lower than the training loss can be explained by regularization being turned on during training like dropout, batch normalization and data augmentation.

5.1.3. Confusion matrix analysis

Figure 5 shows the HMBFNet confusion matrix. The matrix makes it easy to see the true and expected labels. The class name is labeled on each column and the row that corresponds to it. Classes in the dataset are represented by the names of the classes in this study. The exact number of images categorized by that particular model is represented by the diagonal values of matrices.

https://doi.org/10.64823/ijter.2608002

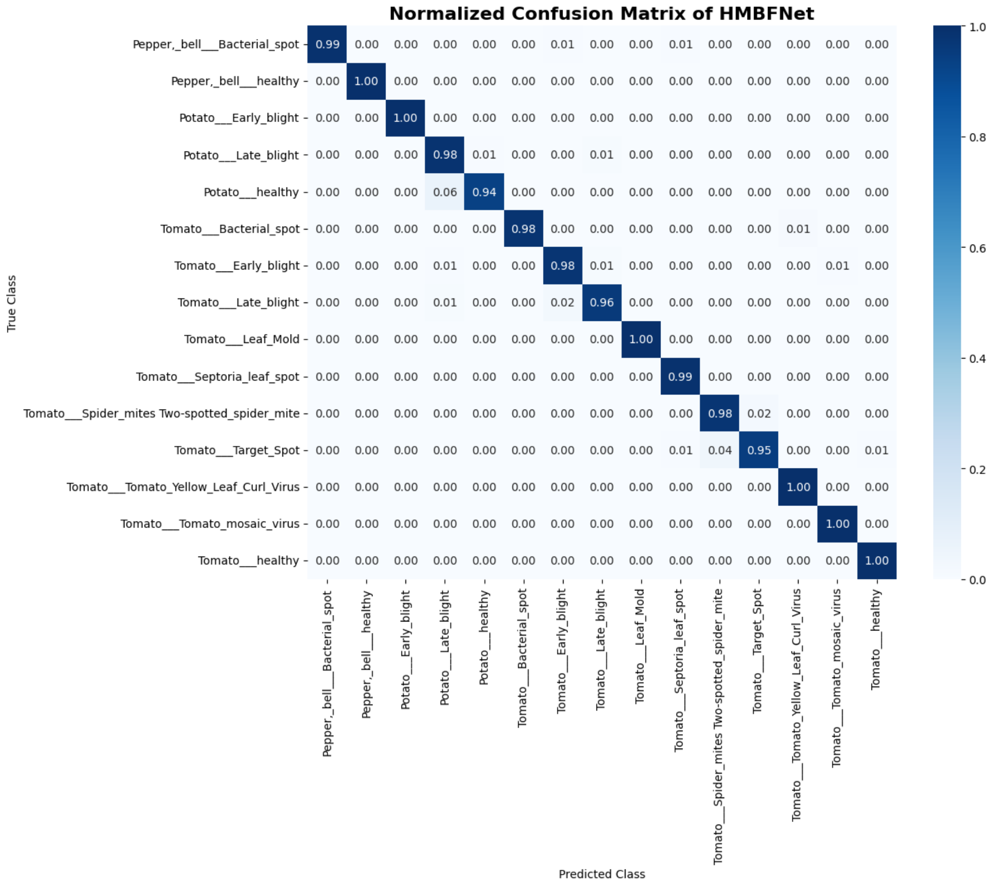


Figure 5: Confusion matrix

5.1.4. Classification report analysis

This section contains an analysis of the classification report that was produced using the confusion matrix displayed in the preceding section. The performance of the suggested model on the provided dataset for each class is examined in this report. The precision, recall, and F1 scores of the model have been displayed in the report. Table 7 displays the detailed analysis of every performance matrix (accuracy score, recall score, F1 score) for every class in the dataset.

Table 7: Classification report of model

Class	Precision	Recall	F1-Score	Support
Pepper, Bell – Bacterial Spot	1.0000	0.9900	0.9950	200
Pepper, Bell – Healthy	1.0000	0.9966	0.9983	295
Potato – Early Blight	0.9950	1.0000	0.9975	200
Potato – Late Blight	0.9610	0.9850	0.9728	200
Potato – Healthy	0.9667	0.9355	0.9508	31
Tomato – Bacterial Spot	0.9976	0.9812	0.9893	425
Tomato – Early Blight	0.9469	0.9800	0.9631	200
Tomato – Late Blight	0.9813	0.9607	0.9709	382
Tomato – Leaf Mold	0.9896	1.0000	0.9948	191
Tomato – Septoria Leaf Spot	0.9915	0.9915	0.9915	354
Tomato – Spider Mites (Two-Spotted Spider Mite)	0.9619	0.9791	0.9704	335
Tomato – Target Spot	0.9708	0.9466	0.9586	281
Tomato – Yellow Leaf Curl Virus	0.9953	0.9981	0.9967	1071
Tomato – Mosaic Virus	0.9867	1.0000	0.9933	74
Tomato – Healthy	0.9937	0.9969	0.9953	318
Overall Accuracy	—	—	0.9862	4557
Macro Average	0.9825	0.9827	0.9826	4557
Weighted Average	0.9863	0.9862	0.9862	4557

With the available settings and simulation parameters, the HMBFNetmodel on the given dataset has performed good in terms of accuracy, recall, prediction and F1 score. Hence, the HMBFNet model can be employed for the system to automate disease detection in plants through analysis of their leaves for early diagnosis of the same.

6. Conclusion

Grain production is being hindered globally by ongoing urbanization, population expansion and fast industrialization, especially in developing countries and urban regions where agricultural yield is in increasing demand. On the other hand, managing agricultural diseases may be a better option than raising crop yields. A team of academics has demonstrated several methods for identifying diseases in rice leaves using a digital picture collection of the leaves. Deep learning has been suggested as a potential remedy for this issue. Plant diseases have thus been classified into multiple classes using a HMBFNetmodel. The architecture of the model aims to achieve four simultaneous goals: memory, speed, accuracy and restricted computational capability. These qualities are determined by the program's features, which necessitate that a diagnosis system be quicker and use less memory than a client-server application because of its more interactive nature. Further, the devised model has been simulated using a publicly available dataset that includes samples from several plant diseases. It was found that the recommended technique is effective in the provided best solution to given context of problem with 98.61% accuracy, precision 98.62, recall and F1-score around 98.61%,

References

[1] Agricultural Statistics at a Glance, 2021. , Government of India, Ministry of Agriculture & Farmers Welfare Directorate of Economics and Statistics https://eands.dacnet.nic.in/PDF/Agricultural%20Statistics%20at%20a%20Glance%20- %202021%20(English%20version).pdf. (January, 2023)

[2] Pradhan, D., & Sharan, P. (2017). Plant disease detection and diagnosis through image processing: A review. *Journal of Agricultural Informatics*, *8(1)*, *1-14*.

[3] Savary, S., Willocquet, L., Pethybridge, S. J., Esker, P., McRoberts, N., & Nelson, A. (2019). The global burden of pathogens and pests on major food crops. *Nature Ecology & Evolution*, 3(3), 430-439.

[4] Sanyal, P. et al. 2007. "Color Texture Analysis of Rice Leaves to Diagnose Deficiency in the Balance of Mineral Levels towards Improvement of Crop Productivity." *Proceedings - 10th International Conference on Information Technology*, *ICIT 2007*: 85–90.

[5] Phadikar, S. and Sil, J., 2008, December. Rice disease identification using pattern recognition techniques. In *2008 11th International Conference on Computer and Information Technology* (pp. 420-423). IEEE.

[6] Yao, Qing Yao Qing et al. 2009. "Application of Support Vector Machine for Detecting Rice Diseases Using Shape and Color Texture Features." *2009 International Conference on Engineering Computation*: 79–83.

[7] Liu, Libo, and Guomin Zhou. 2009. "BP Neural Network." In *2009 International Conference on Computational Intelligence and Software Engineering*, , 9–11.

[8] Sharif, M., Khan, M. A., Iqbal, Z., Azam, M. F., Lali, M. I. U., & Javed, M. Y. (2018). Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. *Computers and Electronics in Agriculture*, *150*, 220–234.

[9] Kerkech, M., Hafiane, A., & Canals, R. (2018). Deep leaning approach with colorimetric spaces and vegetation indices for vine diseases detection in UAV images. *Computers and Electronics in Agriculture*, *155*, 237–243.

[10] Xia, J. A., Cao, H. X., Yang, Y. W., Zhang, W. X., Wan, Q., Xu, L., Ge, D. K., Zhang, W. Y., Ke, Y. Q., & Huang, B. (2019). Detection of waterlogging stress based on hyperspectral images of oilseed rape leaves (Brassica napus L.). *Computers and Electronics in Agriculture*, *159*, 59–68.

[11] Hassanijalilian, O., Igathinathane, C., Doetkott, C., Bajwa, S., Nowatzki, J., & Haji Esmaeili, S. A. (2020). Chlorophyll estimation in soybean leaves infield with smartphone digital imaging and machine learning. *Computers and Electronics in Agriculture*, *174*.

[12] Jiang Feng, Lu Yang, Chen Yu, Cai Di & Li Gongfa (2020), Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computer and Electronics in Agriculture*, *1-9*.

[13] Phadikar, S. and Sil, J., 2008, December. Rice disease identification using pattern recognition techniques. In *2008 11th International Conference on Computer and Information Technology* (pp. 420-423). IEEE.

[14] Abdullah, N.E., Rahim, A.A., Hashim, H. and Kamal, M.M., 2007, December. Classification of rubber tree leaf diseases using multilayer perceptron neural network. In *2007 5th student conference on research and development* (pp. 1-6). IEEE.

[15] Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S. and Batra, N., 2020. PlantDoc: A dataset for visual plant disease detection. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD* (pp. 249-253).

[16] Khamparia, A., Saini, G., Gupta, D., Khanna, A., Tiwari, S. and de Albuquerque, V.H.C., 2020. Seasonal crops disease prediction and classification using deep convolutional encoder network. *Circuits, Systems, and Signal Processing*, *39*, pp.818-836.

[17] Ahmad, I., Hamid, M., Yousaf, S., Shah, S.T. and Ahmad, M.O., 2020. Optimizing pretrained convolutional neural networks for tomato leaf disease detection. *Complexity*, *2020*, pp.1-6.

[18] Li, K., Lin, J., Liu, J. and Zhao, Y., 2020. Using deep learning for Image-Based different degrees of ginkgo leaf disease classification. *Information*, *11*(2), p.95.

[19] Bi, C., Wang, J., Duan, Y., Fu, B., Kang, J.R. and Shi, Y., 2022. MobileNet based apple leaf diseases identification. *Mobile Networks and Applications*, pp.1-9.

[20] Wang, H., Shang, S., Wang, D., He, X., Feng, K. and Zhu, H., 2022. Plant disease detection and classification method based on the optimized lightweight YOLOv5 model. *Agriculture*, *12*(7), p.931.

[21] https://www.kaggle.com/datasets/emmarex/plantdisease.