Proceedings of 15th IOE Graduate Conference
Peer Reviewed
Year: 2024 Month: May Volume: 15
ISSN: 2350-8914 (Online), 2350-8906 (Print)

A Hybrid Approach for Sugarcane Leaf Disease Classification Using CNN and Gradient Boosting Algorithms

Bibek Thapa [a], Smita Adhikari [b], Biplove Pokhrel [c], Ajit Gauli [d]

[a, b, c, d] *Department of Electronics and Computer Engineering, Pashchimanchal Campus, IOE, Tribhuvan University, Nepal*
[a] bibek.thapa619@gmail.com , [b] adsmita1@gmail.com, [c] biplovepokhrel101@gmail.com, [d] ajitgauli111@gmail.com

Abstract
Sugarcane is a globally significant crop, primarily cultivated for sugar and ethanol production. One of the prominent challenges in the sugar industry is the prevalence of sugarcane diseases, leading to substantial financial setbacks for small-scale farmers if not addressed promptly, often resulting in the removal of affected crops. The adoption of deep learning techniques has garnered increasing attention in the realm of research on crop disease classification. In this research, we present an approach to classifying sugarcane leaf diseases utilizing MobileNetV2 with an enlarged receptive field to extract features, coupled with XGBoost, CatBoost, and LightGBM as classifiers. The model is trained and assessed on a dataset comprising sugarcane leaf images categorized into five classes: healthy, mosaic, red rot, rust, and yellow leaf. The proposed MobileNetV2 outperformed baseline MobileNetV2 and other traditional models. When utilizing the proposed MobileNetV2 as the feature extractor, CatBoost acheived better results compared to XGBoost and LightGBM classifiers. The soft voting ensemble of these classifiers yielded the best overall performance.

Keywords
CatBoost, Dilated Convolution, LightGBM, MobileNetV2, Sugarcane Leaf Disease, XGBoost

1. Introduction

Sugarcane is a vital cash crop, playing a crucial role in the global economy and serving as a primary source of sugar production. Nevertheless, numerous challenges hinder the sustainable development of sugarcane, with one major obstacle being the prevalence of various diseases that adversely affect both the quality and production of the crop. The overall health of the sugarcane crop can be significantly compromised by any of these diseases that inflict damage on sugarcane leaves.

The timely identification and precise categorization of sugarcane leaf diseases are crucial for deploying efficient disease control measures. Conventional disease diagnosis methods typically involve visual examination by seasoned agronomists, a process that is often time-consuming, subjective, and susceptible to human mistakes. However, recent progress in computer vision and machine learning methodologies has introduced opportunities for automating the detection and classification of diseases in agriculture, offering a more streamlined and objective approach.

The advancement of DL models in diagnosing plant diseases holds promise for enhancing crop production and minimizing yield losses. The integration of image-processing techniques and ML has notably enhanced the accuracy of plant disease diagnosis[1]. This approach offers potential solutions to challenges faced by both humans and plants. Artificial Intelligence (AI) has made it possible for people to communicate with computers and comprehend their needs. By comparing photos of damaged and healthy foliage, image recognition is essential for recognizing diseased leaves[2]. For this, conventional image processing techniques including segmentation, feature extraction, and categorization have been applied. But DL has become more popular in agricultural research because it can extract more complex feature data than traditional ML algorithms [3].

Transfer learning on pre-trained SOTA models such as VGG-19, Resnet, Mobilenet, etc are proven to excel in classification fields. Hybrid models with the pre-trained model as feature extractor and other ML and DL algorithms as classifiers have also had significant accuracy on image classification. Gradient boosting methods (XGBoost, LightGBM, CatBoost) are commonly used machine learning algorithms and achieve state-of-the-art in some classification tasks. The integration of ensemble methods in deep learning has been a subject of extensive research, garnering significant attention due to their capacity to boost predictive performance by leveraging the diverse strengths of multiple learners. This approach is particularly relevant in the context of image classification tasks, where ensemble models have demonstrated remarkable efficacy in enhancing accuracy and robustness. In the domain of plant leaf disease detection, a transfer learning-based deep ensemble neural network has proven to be more effective than its constituent pre-trained models alone, showcasing the potential of ensemble strategies in handling complex classification tasks with high-dimensional data.

The major contribution of this research are as follows:

- A lightweight MobileNetV2 with increased receptive field on the last inverted residual block using dilated convolution is employed as a backbone feature extractor and gradient boosting methods as classifier.
- The study evaluated the effectiveness of XGBoost, CatBoost, and LightGBM classifiers when employed with the customized MobileNetV2 as the feature extractor. Additionally, a soft voting ensemble technique incorporating these classifiers, yielding improved classification outcomes is introduced.

2. Related Works

The classification of plant leaf diseases has seen extensive exploration with the advancement of deep learning methodologies. Sammy V. Militante and Bobby D. Gerardo [4] employed Convolutional Neural Networks (CNN) to classify sugarcane leaf diseases. The authors gathered a dataset comprising 13,482 images of both healthy and diseased leaves. Their proposed model attained an impressive accuracy of 95% on this dataset. Ramadhani el at.[5] carried out comparison of deep CNN with MobileNetV2 was for the detection of chilli diseases. DeepCNN had an accuracy of 0.8223 where as MobileNetV2 had 0.9151. Mahesh el at.[6], carried out comparison of MobileNetV2 with different pre-trained models on plant village dataset. MobileNetV2 performed better than Resnet50 and InceptionV3. Shanwen Zhang el at.[7] proposed a cucumber leaf disease identification model based on AlexNet. The benefits of global pooling and dilated convolution are combined in the suggested model. Dilated convolution kernel was used in place of the original convolution kernel in AlexNet's Conv1 convolution. Fully connected layers after conv5 was replaced by global pooling layer. The proposed model achieved an accuracy of 94.65 whereas AlexNet had an accuracy of 92.48. Swapnil el at.[8] proposed an ensemble model of CNN and CNN with spatial attention on custom collected dataset. 2569 images of healthy and diseased leaves were categorized into 5 classes. Also comparison of the proposed model with pre-trained models such as VGG19, ResNet50, Xception, MobileNetV2, EfficientnetB7 were carried out. MobileNetV2 achieved the highest accuracy among the pre-trained models.

The proposed model had an accuracy of 0.8653 which outperformed all the SOTA pre-trained models. Kaur el at.[9] proposed a hybrid architecture for leaf disease recognition . VGG-16 and MobileNet were used for transfer learning. Stacking ensemble learning approach was used to merge models. The proposed model performed better than compared pre-trained models.Transfer-learning based deep ensemble was proposed by Vallabhajosyula et al.[10] on plant leaf dataset. The model used ensemble of pre-trained resnet, inceptionv3, nasnet mobile and densenet. The ensemble model demonstrated superior accuracy compared to individual models. Kannan et al. [11] introduced a hybrid approach for detecting rice plant diseases. Their proposed model utilized Inception-ResNetV2 for feature extraction and employed XGBoost ensemble as a classifier. Remarkably, it achieved an accuracy of 99.4% and a precision of 0.965. The preprocessing stage incorporated the use of a Weiner filter. The authors conducted a comprehensive comparison of the model's performance against various machine learning and deep learning algorithms, including SVM, decision tree, KNN, AlexNet, VGG16, and InceptionV3. Their proposed model outperformed all the other models in the comparison.

3. Methodology

3.1 Dataset Preparation

3.1.1 Dataset

Plant village [12] is one of the largest available public dataset on plant leaf disease. But it doesn't include sugarcane plant. This research used the sugarcane leaf disease dataset collected by Daphal and S.M. Koli[8]. Total 2569 RGB images were collected and classified into five different classes. Every photograph was taken in the field from various locations in New Delhi and West Bengal, India.

Table 1: Dataset labels and no. of samples

Class	Number of samples
Healthy	520
Rust	514
Red rot	519
Yellow	505
Mosaic	511

The dataset comprises five distinct classes, namely healthy, rust, red rot, yellow, and mosaic. It is a well-balanced dataset, with each class containing a specific number of images. Specifically, there are 520 images depicting healthy leaves, 514 images showcasing leaves affected by rust, 519 images illustrating leaves affected by red rot, 505 images displaying yellow leaves, and 511 images depicting leaves affected by the mosaic virus.

3.1.2 Dataset Split

We divided the dataset into three separate subsets: the training set, validation set, and test set, to make the creation and evaluation of the machine learning model easier. In the beginning, we used the commonly used 80:10:10 split, dividing the data into three categories: training (80%), validation (10%), and test (10%).

The 80:10:10 split was used as the starting point for training and assessing the model's effectiveness. However, more assessments were carried out utilizing wider test set splits in order to fully evaluate the model's robustness and generalization capacity. Specifically, two different test set splits 70:15:15 and 60:20:20 were used to assess the model's performance.This split highlighted even more how crucial it is to assess the model's performance on a wide variety of previously unseen data, since this will give valuable insights into how well the model generalizes to various contexts.

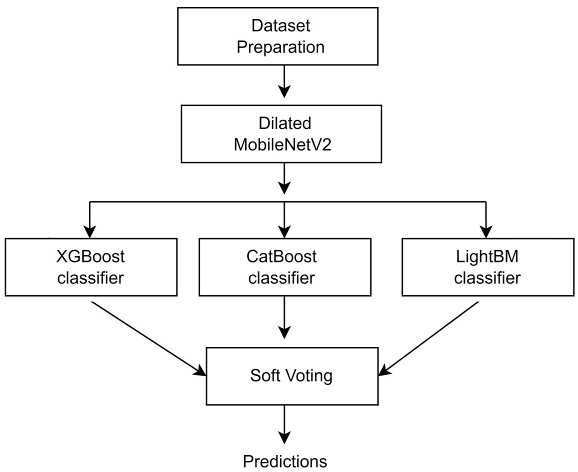

Figure 1: Block diagram of proposed methodology

Figure 2: Dataset Samples

3.1.3 Dataset Preprocessing and Augmentation

All images were resized to a uniform dimension of 224x224 pixels, ensuring consistency across the dataset. Subsequently, a normalization process was applied to the pixel values, typically scaling them to a standardized range. This normalization step is essential for facilitating smoother convergence during the training process.

Apart from resizing and normalization, multiple augmentation approaches were utilized to supplement the training dataset, therefore improving its resilience and diversity. These augmentation methods consist of color jittering, flipping, rotation, scaling, and translation. Flipping involves creating mirror images of the original data by horizontally or vertically flipping the images. Rotation introduces variability in perspective by randomly rotating images by certain angles. Scaling alters the size of images, allowing the model to learn from examples at different distances or zoom levels. Translation shifts images horizontally or vertically, simulating changes in position or viewpoint. Finally, color jittering applies small, random perturbations to the color channels, introducing variations in color and brightness.

3.2 Feature Extrator

The proposed model uses modified MobileNetV2[14] refered as dilated MobileNetV2 for feature extraction. MobileNetV2 is a deep learning model architecture designed for mobile and edge devices. It is a lightweight and efficient neural network architecture developed by Google. MobileNetV2 is an improvement over its predecessor, MobileNet[15], and is specifically optimized for tasks like image classification, object detection, and semantic segmentation in resource-constrained environments.

The inverted residual block are the basic building blocks in MobileNetV2. It consists of an expansion layer, depthwise separable convolution, projection layer, and skip connection. It begins with expanding the input tensor's channels using a 1x1 convolution, followed by a depthwise separable convolution that significantly reduces computational costs

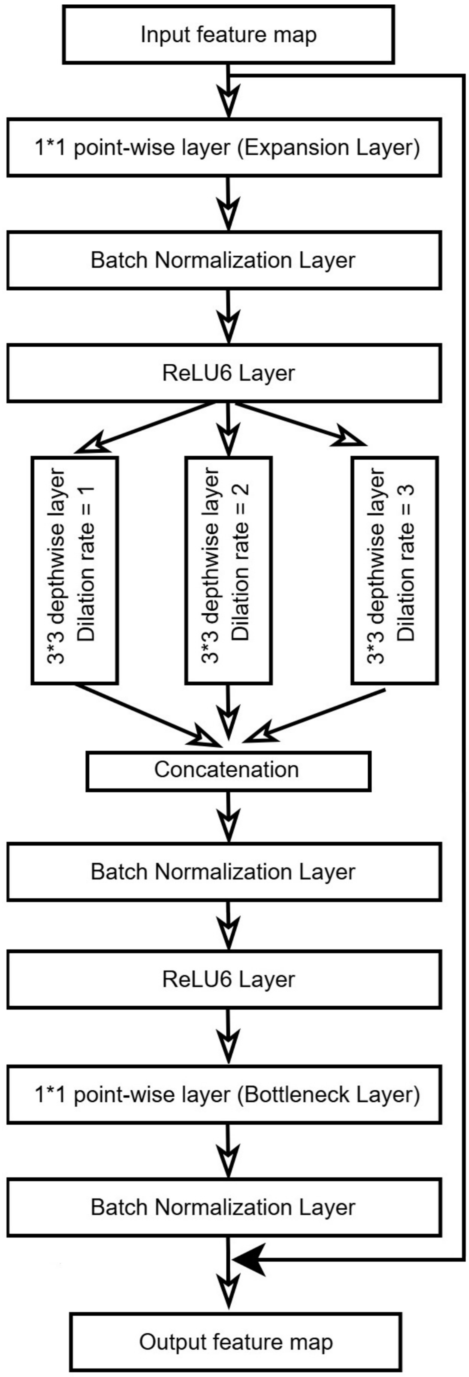

Figure 3: Pyramid Depth-Wise Dilated Separable Convolution Block[13]

while preserving representation capacity. A projection layer also known as bottleneck layer then decreases the dimensionality of the feature maps for computational efficiency. Finally, a skip connection as in Resnet[16] aids in information flow and gradient propagation, facilitating the training of deeper networks.

We replaced the last inverted residual block of MobileNetV2 with Pyramid Depth-Wise Dilated Separable Convolution as proposed in [13]. This block increased the receptive field using concatenation of three 3*3 depthwise dilated layer with dilation rates 1,2 and 3 instead of standard 3*3 depthwise layer in the inverted residual block as shown in Figure 3. Dilated convolution is a convolutional operation with a defined "hole" or spacing between the values in the kernel. This hole is introduced by inserting zeros in the kernel, allowing the convolutional operation to have a larger receptive field without increasing the number of parameters. By adding zeros between the values of the kernel, the spacing, or dilation rate, is increased.

3.3 Classifier

In this research, three distinct gradient boosting classifiers were employed. Gradient boosting methods work by sequentially training a series of weak learners, such as decision trees, in an additive manner. Initially, features extracted from the dilated MobileNetV2 model were individually fed into XGBoost, CatBoost, and LightGBM classifiers. Each classifier operated independently on the extracted features. Following this, soft voting was conducted to combine the predictions from the three classifiers into a final classification decision. In this approach, each classifier generates a probability estimate for every class. These probabilities are then averaged across all classifiers to determine the final probability for each class. Subsequently, the class with the highest calculated probability is selected as the consensus among the ensemble of classifiers.

3.3.1 XGBoost

XGBoost [17], short for eXtreme Gradient Boosting, stands as a robust and efficient machine learning algorithm crafted for supervised learning endeavors. This method extends trees in a level-wise fashion, continuously segmenting features. Each tree identifies the feature and threshold that yield the most significant enhancement in branching and proceeds to execute the split. Essentially, XGBoost surpasses traditional decision trees by employing classification and regression trees as base learners, sequentially amalgamating multiple tree predictions through gradient boosting to minimize errors. Weights play a pivotal role in XGBoost, as they are assigned to all independent variables, which are subsequently input into the decision tree for result prediction. The weight of variables inaccurately predicted by the tree is increased, and these variables are then forwarded to the subsequent decision tree. These individual classifiers then converge to produce a robust and more accurate model.

3.4 CatBoost

CatBoost[18] is a gradient boosting library that is specifically designed for categorical feature support. CatBoost operates as a gradient boosting algorithm, leveraging an ensemble of decision trees to make predictions. What sets CatBoost apart is its specialized handling of categorical features, a feature often challenging for traditional boosting methods. The algorithm employs a technique known as "Ordered Boosting," which involves sorting categorical features by their statistical properties before constructing decision trees. This approach enables CatBoost to create more informed splits, effectively addressing the unique challenges posed by categorical data. During training, CatBoost iteratively builds a sequence of decision trees, with each tree aiming to correct the errors of the ensemble's preceding trees.

3.5 LightGBM

LightGBM [19], also known as Light Gradient Boosting Machine, represents a high-performance gradient boosting framework developed by Microsoft. Renowned for its efficiency, scalability, and speed, LightGBM is particularly well-suited for managing large datasets and intricate machine learning tasks. It belongs to the family of decision tree-based ensemble methods, where numerous weak learners, typically decision trees, are amalgamated to construct a robust predictive model. LightGBM sets itself apart through its distinctive histogram-based learning approach, enabling it to effectively handle categorical features and execute tree construction in a manner that minimizes memory usage. Moreover, LightGBM adopts a leaf-wise growth strategy, expanding the tree by selecting the leaf with the maximum delta loss during each iteration, thereby enhancing its computational speed.

4. Results and Discussion

4.1 Model Training Details

The model was trained on a Kaggle Notebook using a GPU P100 accelerator. Python version 3.10.13 was utilized for implementing the model. Layers 0 through 16 of MobileNetV2 were set to be untrainable, while layers 17 and 18, serving as feature extraction layers, were unfrozen for training. Training made use of the Adam optimizer, and the loss function employed was multi-class cross-entropy. We tuned the settings of the model by experimenting with different combinations of hyperparameters. We adjusted the learning rate to 0.00001 and batch sizes of 32 and 64. Additionally, we varied the dropout rates between 0.25 and 0.5 to optimize the model's performance. The model was trained with a patience of 3 epochs to prevent overfitting. The hyperparameter tuning was conducted using an 80:10:10 dataset split for training, validation, and testing, respectively. We also conducted k-fold validation for our model. Initially, we split the dataset into 90% for training and 10% for testing unseen data. Then, within the training data, we divided it into 9 folds. For each iteration, 8 folds were used for training the model, while the remaining fold was used for validation. We utilized the model with the most effective hyperparameters to extract features, which were then inputted into XGBoost, CatBoost, and LightGBM classifiers.

4.2 Results

Table 2 presents the results obtained from training the model using different sets of hyperparameters. Notably, the model trained with a batch size of 32, dropout rate of 0.5, and a learning rate of 0.00001 achieved the highest accuracy. Since the dataset was balanced, accuracy was chosen as the main metric for assessing the model's performance.

Table 3 shows the results of k-fold cross-validation process involved iteratively training and validating the model over 9 folds. Throughout the iterations, validation accuracies ranged from 0.956 to 0.972, showcasing the model's performance variability across different subsets of the data. Test accuracies, on the other hand, fluctuated between 0.968 and 0.98,

Table 2: Experiment Results on dilated MobileNetV2

Batch Size	Dropout	Accuracy	Precision	Recall	F1 Score
32	0.25	0.9722	0.973	0.9723	0.9724
32	0.5	**0.9762**	0.9788	0.9753	0.9760
64	0.25	0.9762	0.9764	0.9762	0.9762
64	0.5	0.9722	0.9734	0.9715	0.9719

indicating the model's effectiveness in generalizing to unseen data.

Table 3: K-fold cross validation results on dilated MobileNetV2

Iteration	Validation Accuracy	Test Accuracy
1	0.964	0.968
2	0.968	0.98
3	0.956	0.976
4	0.964	0.976
5	0.956	0.98
6	0.964	0.98
7	0.964	0.968
8	0.968	0.968
9	0.972	0.976

We used the features extracted from the dilated MobileNetV2 model, specifically focusing on the output of its average pooling layer, as inputs for training three different classifiers: XGBoost, CatBoost, and LightGBM. Additionally, we performed soft voting, combining predictions from these classifiers to make final decisions. Table 4 offers a comprehensive overview of the performance metrics for three different classifiers: XGBoost, CatBoost, LightGBM, and soft voting ensemble of these classifiers. Each classifier's accuracy, precision, recall, and F1 score are listed, providing valuable insights into their effectiveness in classifying instances within the dataset.

Table 4: Performance of Dilated MobileNetV2 with different classifiers

Classifier	Accuracy	Precision	Recall	F1 Score
XGBoost	0.9762	0.9774	0.9753	0.9757
CatBoost	0.9802	0.9816	0.9793	0.9799
LightGBM	0.9563	0.9541	0.9577	0.9548
Soft Voting	0.9802	0.9811	0.9793	0.9796

Among the individual classifiers injunction with dilated MobileNetV2, CatBoost exhibits the highest accuracy of 0.9802, closely followed by Soft Voting, which achieves the same accuracy. This suggests that CatBoost performs exceptionally well as a standalone classifier. XGBoost also delivers commendable accuracy at 0.9762, while LightGBM achieves a slightly lower accuracy of 0.9563.

We also evaluated the model on larger portions of the test set, specifically using splits of 70:15:15 and 60:20:20. The performance of the models on various dataset split is shown in Table 5 Among the individual classifiers, CatBoost showed the best performance when paired with the dilated MobileNetV2 feature extractor, followed by XGBoost and LightGBM on each split. The soft voting ensemble produced similar results as the CatBoost classifier in the 80:10:10 split but performed better than all the classifiers on 70:15:15 and 60:20:20 splits.

Table 5: Accuracy of the classifiers using dilated MobileNetV2 feature extractor on different dataset split

Dataset Split	FCN	XGBoost	Catboost	LightGBM	Soft Voting
80:10:10	0.9762	0.9762	**0.9802**	0.9563	**0.9802**
70:15:15	0.9550	0.9629	0.9656	0.9576	**0.9682**
60:20:20	0.9623	0.9642	0.9662	0.9603	**0.9682**

Table 6: Performance comparison of different models

Model	Accuracy
MobileNetV2	0.9365
Dilated MobileNetV2	0.9762
Proposed Model	**0.9802**
CNN + CNN with spatial attention[8]	0.8653
AlexNet[20]	0.8750
DarkNet-53[20]	0.8730
GoogLeNet[20]	0.8750
ResNet50[20]	0.8570
VGG-19[20]	0.9220

4.3 Comparison with other models

In addition to training the proposed dilated MobileNetV2 model, we also trained a baseline pretrained MobileNetV2 using the same settings. The MobileNetV2 model achieved an accuracy of 0.9365. We also included models from other authors for comparison. The accuracy scores of various models in sugarcane leaf disease classification are summarized in Table 6. The proposed model outperformed the work conducted by Daphal et al. [8], achieving superior results. The authors employed a stacking ensemble comprising two networks: a sequential CNN and a CNN incorporating spatial attention, resulting in an accuracy of 0.8635. Among these models, the proposed model achieves the highest accuracy of 0.9802, surpassing all others.

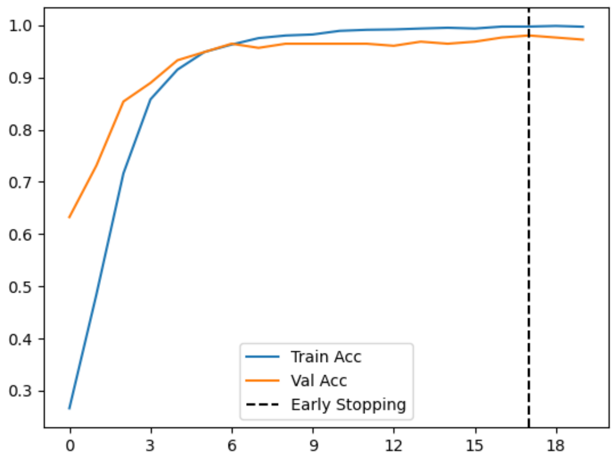

Figure 4: Accuracy curve for dilated MobileNetV2

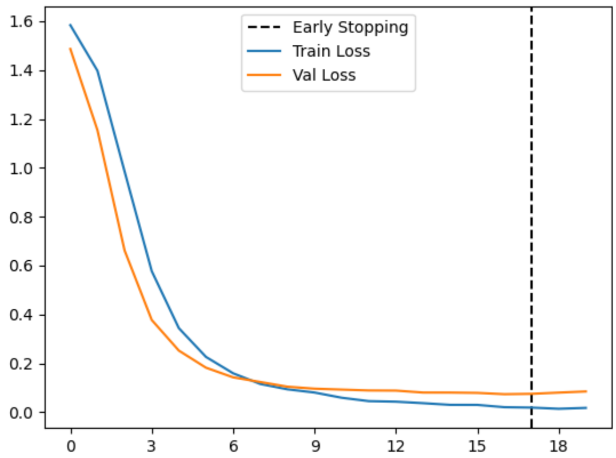

Figure 5: Loss curve for dilated MobileNetV2

Figure 6: Confusion matrix for dilated MobileNetV2

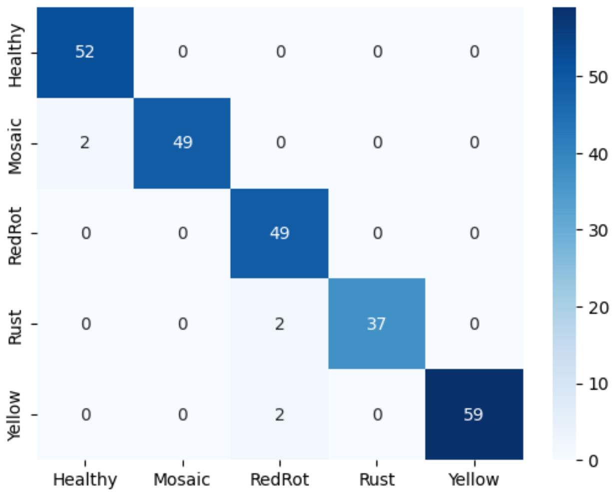

Figure 7: Confusion matrix for XGBoost classifier

Figure 8: Confusion matrix for Catboost classifier

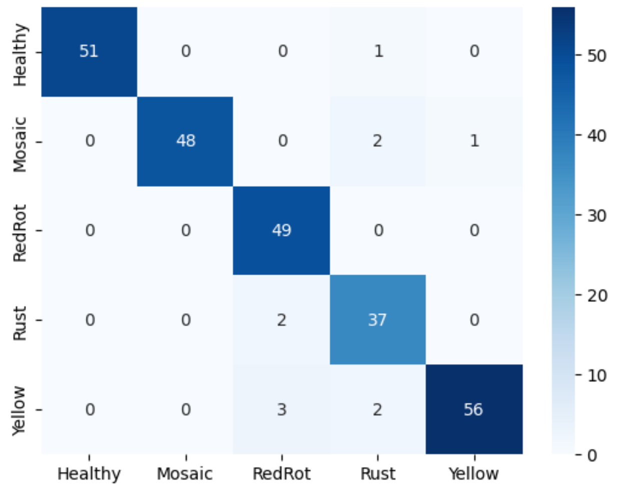

Figure 9: Confusion matrix for LightGBM classifier

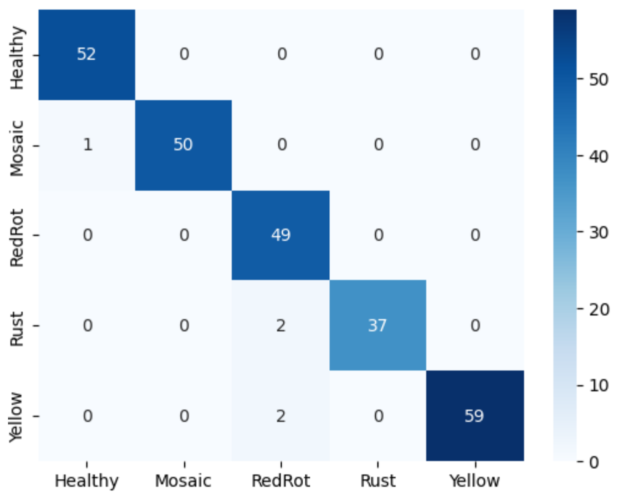

Figure 10: Confusion matrix for soft voting

5. Conclusion

Our research presents a comprehensive approach for sugarcane leaf disease classification using MobileNetV2 with dilated depthwise convolution as a feature extraction method and XGBoost, CatBoost, and LightGBM as classifiers. Increase in receptive field of MobileNetV2 in the last residual block can enhance the feature extraction capability of the model. Through extensive experimentation and evaluation on a diverse dataset, we have demonstrated the effectiveness of our proposed model in accurately identifying sugarcane leaf diseases.

The achieved accuracy of 98.02% surpasses the stacking ensemble of sequential CNN and CNN with spatial attention [8], baseline MobileNetV2 and traditional models such as AlexNet, DarkNet-53, GoogLeNet, ResNet50, and VGG-19. In the 80:10:10 split of the dataset, CatBoost, when utilized with dilated MobileNetV2 as feature extraction, yielded the most optimal outcome, matching the performance achieved by the soft voting ensemble of XGBoost, CatBoost, and LightGBM classifiers. However, our analysis of various dataset splits indicates that the soft voting ensemble of probabilities derived from XGBoost, CatBoost, and LightGBM classifiers can notably enhance classification accuracy.

In order to further enhance classification performance, future research projects might concentrate on enlarging the dataset to encompass a wider variety of sugarcane leaf diseases and investigating different deep learning architectures and ensemble methodologies.

References

[1] R. I. Hasan, S. M. Yusuf, and L. Alzubaidi. Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. *Plants*, 9(10):1302, Oct. 01 2020.

[2] J. Blumenthal, D. B. Megherbi, and R. Lussier. Supervised machine learning via hidden markov models for accurate classification of plant stress levels & types based on imaged chlorophyll fluorescence profiles & their rate of change in time. In *2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*. IEEE, Jun. 2017.

[3] M. Chen et al. Three-dimensional perception of orchard banana central stock enhanced by adaptive multi-vision technology. *Computers and Electronics in Agriculture*, 174:105508, Jul. 2020.

[4] V. Militante and B. D. Gerardo. Detecting sugarcane diseases through adaptive deep learning models. *Journal of Critical Reviews*, 07(15), 2021.

[5] A. N. Ramadhani, G. W. Saraswati, R. T. Agung, and H. A. Santoso. Performance comparison of convolutional neural network and mobilenetv2 for chili diseases classification. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 7(4):940–946, Aug. 12 2023.

[6] M. TR, R. Sivakami, I. Manimozhi, N. Krishnamoorthy, and B. Swapna. Early predictive model for detection of plant leaf diseases using mobilenetv2 architecture. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2):46–54, Feb. 17 2023.

[7] S. Zhang, S. Zhang, C. Zhang, X. Wang, and Y. Shi. Cucumber leaf disease identification with global pooling dilated convolutional neural network. *Computers and Electronics in Agriculture*, 162:422–430, Jul. 2019.

[8] S. D. Daphal and S. M. Koli. Enhancing sugarcane disease classification with ensemble deep learning: A comparative study with transfer learning techniques. *Heliyon*, 9(8):e18261, Aug. 2023.

[9] P. Kaur et al. Recognition of leaf disease using hybrid convolutional neural network by applying feature reduction. *Sensors*, 22(2):575, Jan. 12 2022.

[10] S. Vallabhajosyula, V. Sistla, and V. K. K. Kolli. Transfer learning-based deep ensemble neural network for plant leaf disease detection. *Journal of Plant Diseases and Protection*, 129(3):545–558, May 02 2021.

[11] E. Kannan. An efficient deep neural network for disease detection in rice plant using xgboost ensemble learning framework. *International Journal of Intelligent Systems and Applications in Engineering*, 10(3):116–128, Oct. 1 2022.

[12] Kaggle. Plantvillage dataset. `https://www.kaggle.com/datasets/emmarex/plantdisease`, accessed 2024.

[13] H. O. A. N. H. Nguyen. A lightweight and efficient deep convolutional neural network based on depthwise dilated separable convolution. *Journal of Theoretical and Applied Information Technology*, 98(15):2937–2947, 2020.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[15] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[17] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 785–794, 2016.

[18] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin. Catboost: Unbiased boosting with categorical features. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[19] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 3149–3157. 2017.

[20] Muhammed Yıldırım Yunus Emre Karaca, Serpil Aslan. Deep learning architectures performance in plant leaf diseases. *All Sciences Proceedings*, 2023.