


Automatic plant disease detection using computationally efficient convolutional neural network

Muhammad Rizwan¹ | Samina Bibi¹ | Sana Ul Haq¹ | Muhammad Asif¹ |
Tariqullah Jan² | Mohammad Haseeb Zafar³ 

¹Department of Electronics, University of Peshawar, Peshawar, Pakistan

²Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan

³Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff, UK

Correspondence

Mohammad Haseeb Zafar, Cardiff School of Technologies, Cardiff Metropolitan University, Cardiff CF5 2YB, UK.

Email: mhzafar@cardiffmet.ac.uk

Abstract

Agricultural plants are the fundamental source of nutrients worldwide. The attack of diseases on these plants leads to food scarcity and results in a catastrophic situation. These diseases can be prevented by using manual or automatic approaches. The manual approach, where plant pathologists inspect fields, is costly, error-prone, and time-consuming. Alternatively, automatic approaches utilize 2D plant images processed through machine learning. The current study opts for the later approach due to its advantages in terms of speed, efficiency, and convenience. Convolutional neural network (CNN)-based prominent models, such as MobileNet, ResNet50, Inception, and Xception, are preferred for automatic plant disease detection due to their high performance, but they demand substantial computational resources, limiting their use to a class of large-scale farmers. The proposed study developed a novel CNN model that is suitable for small-scale farmers. The numerical outcomes indicate that the proposed model surpassed the state-of-the-art models by achieving an average accuracy of 96.86%. The proposed model utilized comparatively limited computational resources as analyzed through floating-point operations (FLOPs), number of parameters, computation time, and model's size. Furthermore, a statistical approach was proposed to analyze a model while collectively accounting for its performance and computational complexity. It is observed from the results that the proposed model outperformed the state-of-the-art techniques in terms of both average recognition accuracy and computational complexity.

KEYWORDS

automatic plant disease detection, computational complexity, collective data analysis, group CNN, separable CNN

1 | INTRODUCTION

Food crops serve as the very foundation of human sustenance, supplying the vital nourishment that energizes both our bodies and intellects. The cultivation and ample availability of these crops stand as critical pillars upholding global food

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Author(s). *Engineering Reports* published by John Wiley & Sons Ltd.

security and the overall health of our planet. Scarcity of crops has historically led to devastating disasters, with instances in Europe, Bengal, and the USA serving as stark reminders of the profound consequences that can ensue.¹ One of the main causes of crop loss is infectious diseases caused by viruses, bacteria, oomycetes, fungi, and nematodes. There are two main approaches to prevent such catastrophic situations: manual and automatic. In the manual approach, plant pathologists visit the farm fields for careful analysis and treatment of the affected plants. This approach is expensive, time-consuming, and impractical because it is difficult to cover all fields in the country.¹ The automatic approach, on the other hand, includes the utilization of advanced machine learning and deep learning techniques. This approach acquires the image data of the plant leaves, extracts the key features, and analyze it using the given models for disease classification. The prominent approaches introduced for the automatic classification of plant diseases include Support Vector Machine (SVM), Random Forest, k-Nearest Neighbor (k-NN), Naive Bayes, and CNN.² The CNN model exhibited superior performance over the others in this domain. Despite its better feature extraction capabilities, the CNN architecture still faces certain limitations owing to the varying leaf sizes, complex backgrounds, illumination effects, and computational cost of the model. The use of high-performance models with low computational complexity has the added benefits of reduction in hardware and energy requirements and thus making them more accessible and cost-effective for small-scale farmers.

The proposed study developed a small-scale farmer's model using CNN that outperformed other state-of-the-art models by attaining a comparatively higher average accuracy with less computational complexity. Furthermore, we proposed a statistical approach to collectively analyze the models while accounting its comparative performance (i.e., average accuracy) and computational complexity (i.e., FLOPs, number of parameters, computation time, and model's size). This approach quantifies the extent to which a model outperforms others. In addition, the proposed model's significance was appraised by comparing it to several top-tier models, that is, MobileNet V2,³ ResNet50 V2,⁴ Inception V3,⁵ Xception,⁶ and a 9-layer deep CNN.⁷

The paper's organization is as follows: Section 2 provides a literature review concerning the automatic detection of plant diseases. Sections 3 and 4 elaborate on the dataset and preprocessing techniques employed prior to training the CNN models. The architectural design of the proposed model is discussed in Section 5, while Section 6 describes the state-of-the-art models utilized for comparison. Sections 7 and 8 discuss the experimental setup and evaluation metrics used to assess the performance and computational complexity of the models. Sections 9 and 10 present the results and discussion, which are subsequently followed by the conclusion.

2 | RELATED WORK

The automatic detection of plant diseases using CNN was first performed by Sladojevic et al.⁸ The proposed model was fine-tuned on CaffeNet,⁹ that is, a variant of Alex Net,¹⁰ using a self-collected dataset of 15 plant disease classes. The model attained the best average accuracy of 96.30%. Sardogan et al.¹¹ developed a CNN architecture with a consistent number of input channels. The proposed model utilized the Linear Vector Quantization (LVQ) algorithm for fully connected layers. The model underwent training using the PlantVillage dataset,¹² focusing on 5 plant disease classes, and attained an average accuracy rate of 86.00%. Frentinos¹³ fine-tuned various CNN models, namely VGG16,¹⁴ AlexNet,¹⁰ and OverFeat,¹⁵ on a plant disease dataset of 87,848 samples comprising 58 plant disease classes. The proposed study revealed that VGG16¹⁴ achieved the best average accuracy of 99.53%.

Kamal et al.¹⁶ developed two variants of MobileNet,¹⁷ namely Modified MobileNet and Reduced MobileNet. The models featured retracting the final five convolution layers from MobileNet and introducing a width multiplier to control the channel size. The PlantVillage dataset¹² training was used for the model, encompassing 54 distinct classes of plant diseases. The average accuracy of Modified MobileNet was 97.67%, and that of Reduced MobileNet was 98.34%. The MobileNet¹⁷ achieved an average accuracy of 98.65%. The Modified MobileNet and Reduced MobileNet, slightly lagged MobileNet¹⁷ in terms of average accuracy but were found to lead with a comparatively smaller number of parameters. Geetharamani and Pandian⁷ developed a 9-layer CNN model. The model consisted of three convolution blocks, where rather than increasing the channel size in the deeper layer, the channel size was reduced. The model underwent training with the PlantVillage dataset,¹² encompassing 39 different plant disease classes, and achieved an average accuracy of 96.46%.

Sagar and Jacob¹⁸ utilized a transfer learning approach for five prominent models, namely VGG16,¹⁴ Inception,¹⁹ ResNet50,²⁰ InceptionResNet,²¹ and DenseNet169.²² The proposed approach fine-tuned the top layers, that is, additional convolution, fully connected, and classification layers. Each of the five models was trained using the PlantVillage dataset¹² for 38 plant disease classes. The ResNet50 achieved a notably elevated average accuracy rate of 94.00% in comparison.

Abattah et al.²³ introduced a customized approach for plant disease detection by utilizing a base network of DenseNet-77²² in conjunction with a custom CenterNet.²⁴ DenseNet was employed for feature extraction from the input samples, and these features were subsequently utilized in the training of the CenterNet. The PlantVillage dataset¹² was utilized to train the proposed model on 39 plant disease classes. The proposed model achieved an average accuracy of 99.98%.

In addition to deep learning, utilization of multimodal approaches such as CLIP,²⁵ M-FLAG,²⁶ and Med-uniC,²⁷ have an advantage of precision diagnosis, leading to improved accuracy of detection. The capability of CLIP²⁵ to determine images and text can be employed in cases of visual symptoms associated with textual expressions on the symptoms of plants, while M-FLAG²⁶ establishes a spatial and semantic relationship between modalities for providing the contextual insights. Med-uniC's²⁷ ability to carry out image analysis, specially made for medical images, can be turned around to determine plant diseases based on their visible symptoms. One of the main drawbacks includes the specificity of Med-uniC's²⁷ training data and its role in generalization. That means generation of practical solutions might be restricted due to a lack of diverse plant diseases knowledge. Despite the significant contribution of multimodal methods, the CNN approach still dominates the area of plant disease identification owing to its sophisticated architectures intended for image recognition.

Most research work proposed so far has either developed the model with remarkable performance and high computation complexity or vice versa. This may lead to the conclusion that there is a trade-off between model's performance and computational complexity. To address this gap, the following research contributions are made:

1. A cutting-edge CNN model suitable for small-scale farmers was developed for automatic detection of plant diseases.
2. A statistical approach was developed to analyze the models while accounting both the models' performance and computational complexity.
3. A detailed analysis of the state-of-the-art models' performance, computational complexity, and significance was conducted to ensure the proposed model's applicability in real-time applications.

3 | DATASET

A large amount of data is required to train deep learning models effectively. For appropriate training of the models, the dataset should account for some of the visual features, including the illumination effect and variations in leaf size, shape, color, and position. While considering these features, we explored several datasets, that is, DiaMOS,²⁸ PlantDoc,²⁹ PDD271,³⁰ and PlantVillage.¹² The DiaMOS and PlantDoc datasets have limited number of samples and classes, while the PDD271 and PlantVillage datasets contain a comparatively larger number of samples and classes. However, the PDD271 dataset is not publicly available, which limits its accessibility to researchers and practitioners. The PlantVillage dataset, on the other hand, serves the required purpose as it is the largest publicly available dataset with 55,448 data samples of 39 plant disease classes.

In this study, the PlantVillage dataset is used for the analysis. The distribution of the dataset across the classes is depicted in Figure 1. The samples in the dataset were collected by plucking the leaves of the plants and placing them on gray or black paper to prevent a complex background, as shown in Figure 2.

4 | PREPROCESSING

The PlantVillage dataset contains sufficient samples, but to prevent the deep learning models from overfitting, various image transformation approaches⁷ were utilized, that is, flipping, rotating, principal component analysis (PCA), color augmentation, gamma correction, scaling, and noise injection. These transformation approaches increased the data size from 55,448 to 61,486, thereby enhancing the generalization of the models. Moreover, data augmentation enhances the resilience and trustworthiness of models by introducing numerous variations to each sample, substantially enriching the input representations.^{31,32} Figure 3 depicts the sample distribution of the augmented dataset.

5 | PROPOSED MODEL

In the proposed research, a CNN model was developed using a standard convolution approach. The proposed model was then further improved using the CNN variants to enhance its performance and reduce the

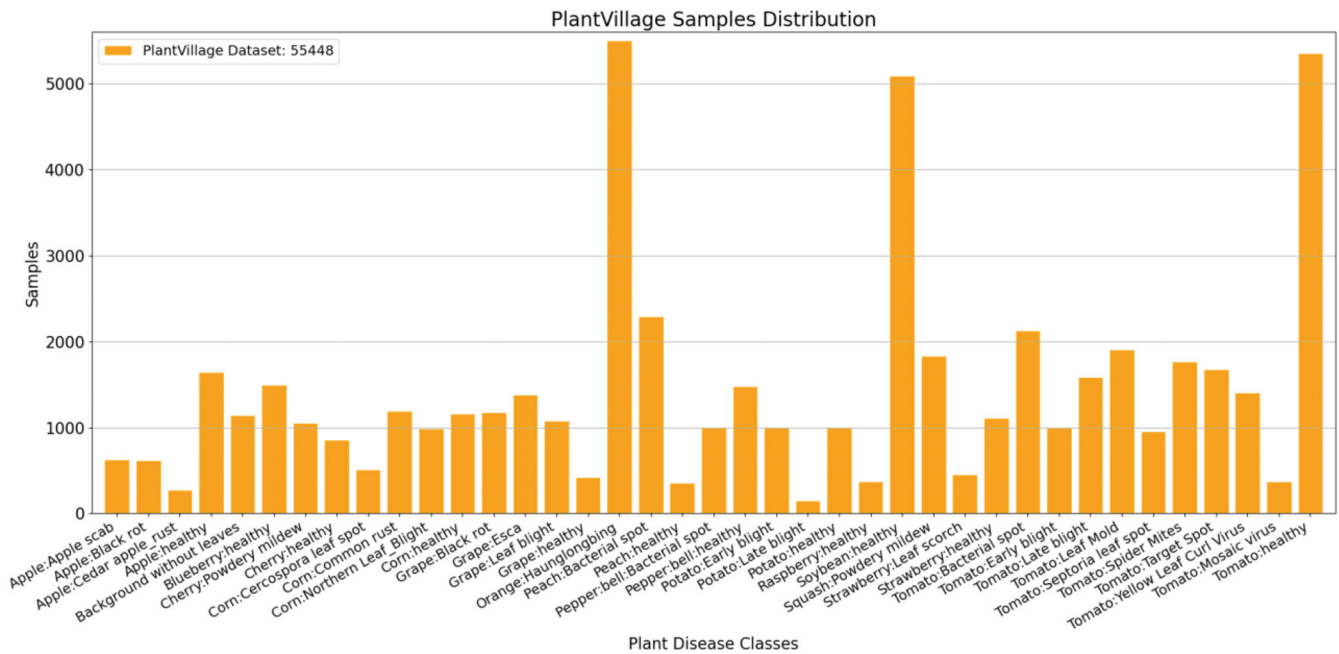


FIGURE 1 Distribution of plant disease classes in the PlantVillage dataset.



FIGURE 2 Data samples from PlantVillage dataset.

computational complexity. The subsequent subsections offer an extensive exploration of the proposed model and various CNN variants.

5.1 | Standard CNN

The proposed standard CNN model is based on the conventional convolution approach.⁷ The model consists of three convolution blocks and a classification block, as illustrated in Figure 4A. The architectural design of all three convolutional blocks is identical to that of the first convolution block. The following outlines the architectural characteristics of the proposed model:

1. The convolutional blocks employ a 3×3 kernel to minimize the weight count, accompanied by a stride of 1.
2. After each convolution layer, maximum pooling is applied for the extraction of key features and downsampling.

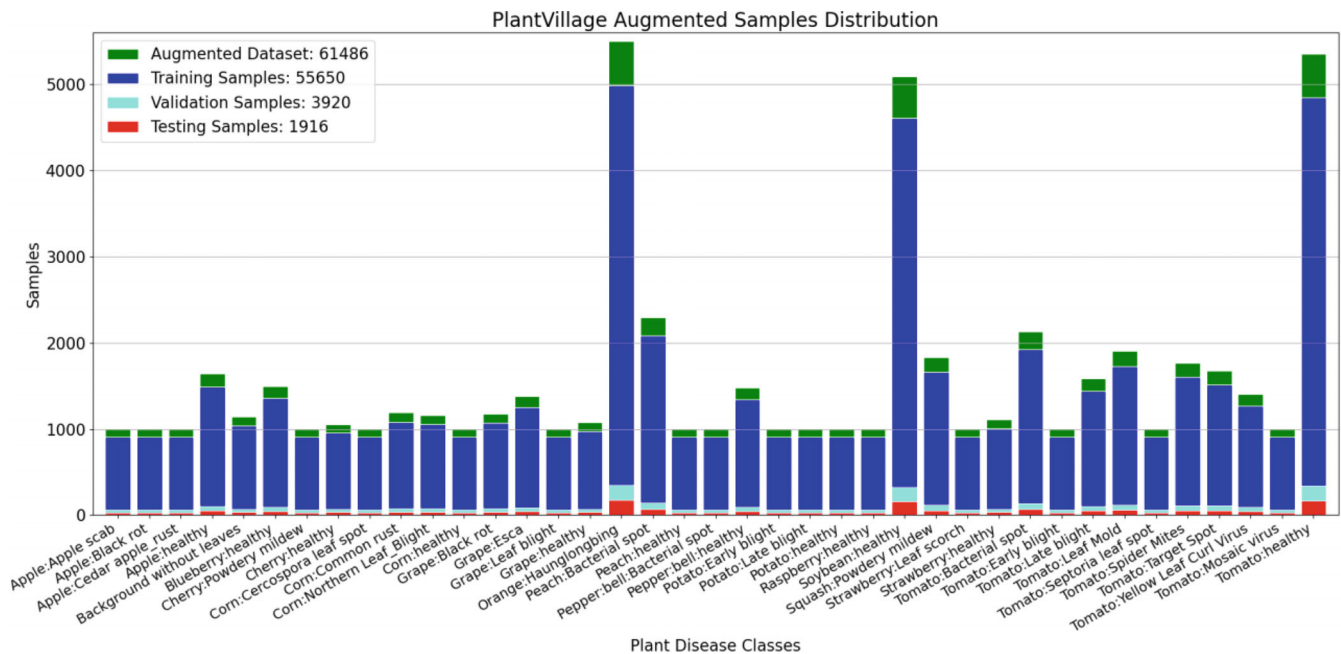


FIGURE 3 Distribution of plant disease classes in the augmented PlantVillage dataset.

3. The classification block used one fully connected layer comprising 128 nodes, followed by a 0.2 dropout to prevent the model from overfitting.

Detailed analysis and evaluation of the proposed model are presented in Section 9.

5.2 | Depth-wise separable CNN

This approach is extremely effective since it greatly reduces the model's computational complexity by reducing the number of parameters. Instead of performing the spatial and channel-wise convolutions in one step, they are performed separately. In deep learning, the spatial and channel-wise convolutions are termed as depth-wise and point-wise convolutions, respectively. The proposed depth-wise separable CNN architecture is shown in Figure 4B. The model exhibits a similar architectural design as that of the proposed standard CNN shown in Figure 4A, has the following key features:

1. Convolution block 1 uses the conventional convolution approach.
2. The convolution blocks 2 and 3 use depth-wise separable convolution to reduce the model's complexity.
3. The classification block exhibits feature similar to those of the standard CNN approach.

5.3 | Group CNN

The group CNN³³ is another novel approach that reduces the computational complexity of the model. In this approach, the convolution operations are performed in groups, where each group has the same spatial dimension but the channel dimension, C_{in} , is divided by defined groups. In this research, we have used four groups as it was observed to result in better model's performance.

The proposed standard CNN model was modified using the group convolution algorithm, as shown in Figure 4C. The key features of the proposed GCNN model are as follows:

1. Convolution block 1 uses the standard convolution approach.
2. The convolution blocks 2 and 3 implement the group convolution in four different groups.
3. The classification block comprises features that are similar to those of the standard CNN model.

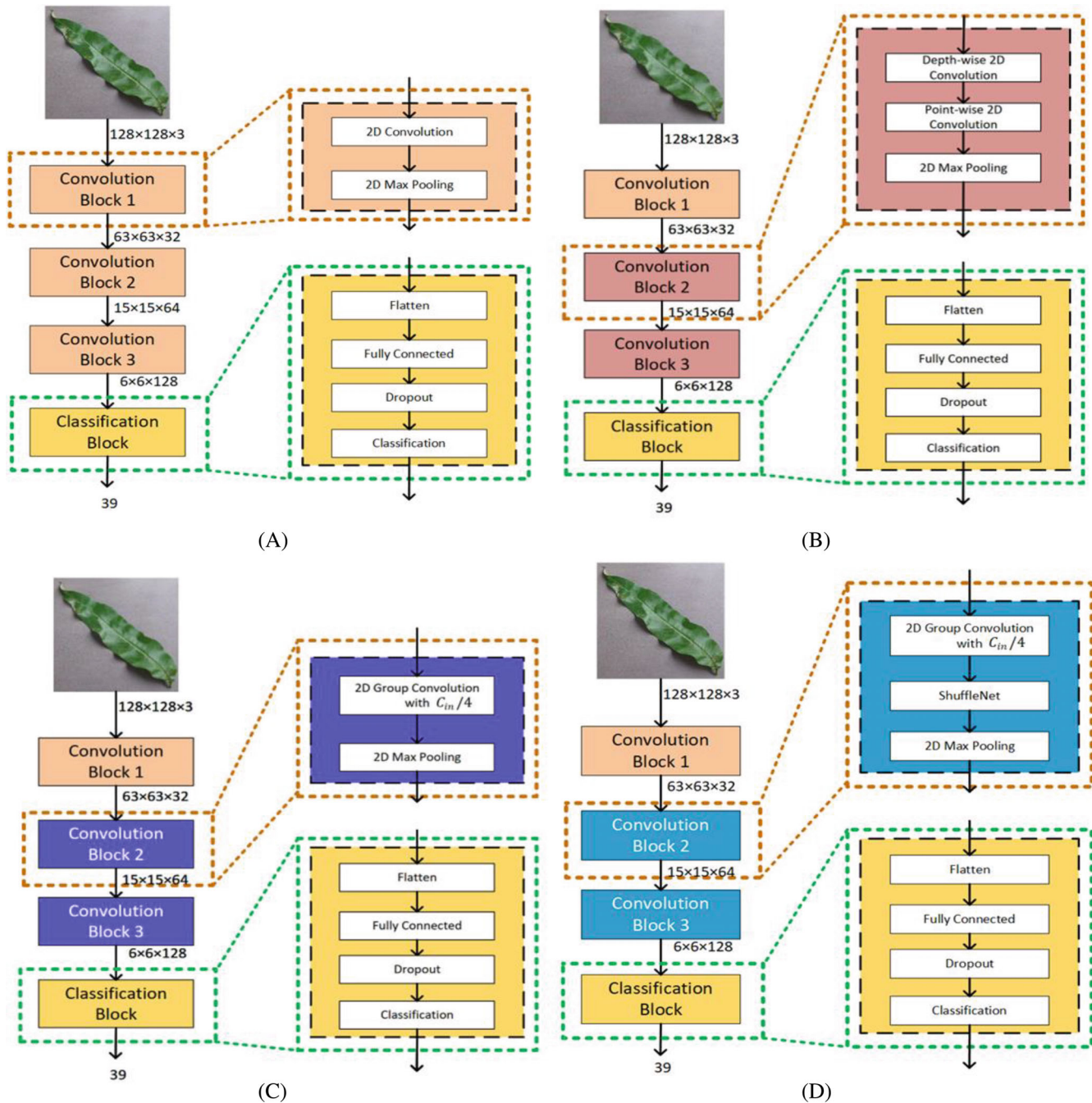


FIGURE 4 Proposed model architecture design using (A) standard CNN, (B) separable CNN, (C) GCNN, and (D) ShuffleNet.

5.4 | ShuffleNet

The GCNN model suffers due to restricting the extracted features to a certain group, which affects the model's performance. The information flow between different groups is blocked, which weakens the feature extraction while training the model. This problem was overcome by introducing ShuffleNet.³³ The ShuffleNet shuffles the channel-wise features of the defined groups with one another. This approach improves the recognition accuracy while sustaining the computational complexity of the GCNN model. Figure 4D depicts the architectural advancements made to convolution blocks 2 and 3 of the proposed ShuffleNet model.

The performance evaluation of the proposed Standard CNN, depth-wise separable CNN, GCNN and ShuffleNet models is discussed in Section 9.

6 | COMPARATIVE MODELS

The evaluation of the proposed model's performance and computational complexity was carried out against five top-tier models, that is, MobileNet V2,³ ResNet50 V2,⁴ Inception V3,⁵ Xception,⁶ and 9-layer deep CNN.⁷ The 9-layer-deep CNN, among the rest, was specifically designed for plant disease detection. The model was computationally less complex and was evaluated using a parameterized rectified linear unit (PReLU). The pre-trained models, that is, MobileNet V2,³ ResNet50 V2,⁴ Inception V3,⁵ and Xception,⁶ were fine-tuned by freezing the convolution base and replacing the classification block with a configuration similar to that of the proposed model.

7 | MODELS' SCALABILITY AND ADAPTABILITY

CNN models show great promise in recognizing plant diseases solely from leaf images, yet they face limitations that hinder their scalability in this domain. These limitations stem from factors such as input image resolution, dataset size, data augmentation, and model's architecture. In the subsequent sections, we delve into a detailed discussion of each of these factors to elucidate their impact on the scalability and deployment of CNN models in the realm of automatic plant disease detection.

7.1 | Input image resolution

The quality of the input image significantly influences the performance of the model. Lower-resolution input data can pose challenges for the model in extracting essential details, potentially resulting in inaccurate predictions. Hence, it is imperative to utilize higher input resolutions for CNN models to ensure optimal performance.³⁴

7.2 | Dataset size

The deep learning model requires a massive dataset, as mentioned previously in Section 3. This is because the more versatile dataset results in better training of the model and greater generalization. It is evident from the literature that the model with better generalization has greater scaling capabilities.³⁵

7.3 | Data augmentation

The data augmentation is one of the most significant factors in terms of the scalability of CNNs, as it allows to simply handle smaller datasets without negatively impacting the model's performance. Through the enhancement of the diversity and number of data points, data augmentation permits the neural network with many layers to have better generalization to the unseen samples, leading to a reduction of the risk of overfitting and improving the scalability. Enriched with the added data, CNNs have the ability to learn more arbitrary and general features leading to their utilization in a wider variety of inputs and situations. This scalability would allow CNNs to exhibit the same level of efficiency even when trained on increasingly more substantial and intricate datasets, rendering them ideal for utilization across various domains.

7.4 | Model's architecture

The architectural design of CNN models significantly impacts their suitability for real-time applications. Models with higher computational complexity may be challenging to deploy, especially on limited-resource devices. Factors such as the width, depth, and arrangement of the architecture play a crucial role in determining the scalability of the model while considering its computational requirements. Therefore, optimization of these architectural elements is essential to ensure that CNN models remain efficient and practical for deployment in real-world scenarios, particularly on devices with constrained resources.

8 | EXPERIMENTAL SETUP

The proposed study utilized the augmented PlantVillage dataset, as discussed in Section 4. The dataset was divided into three segments: training, validation, and testing. The dataset encompassed a total of 61,486 samples, with 55,650 allocated for training, 3920 for validation, and 1916 for testing purposes. While training the model, the batch loss was calculated using the categorical cross-entropy function formulated in (1).

$$\text{Loss}_{\text{categorical}} = - \sum_i t_i \log(y_i). \quad (1)$$

Here, t_i represents the actual outputs, and y_i represents the predicted outputs in the context of categorical loss.

The loss for each of the proposed models and the 9-layer deep CNN⁷ was optimized using the Stochastic Gradient Descent (SGD).³⁵ The SGD was configured with a learning rate ranging from 10^{-3} to 10^{-6} and a momentum of 10^{-5} . The loss for the remaining models was optimized using the ADAM optimizer,³⁶ as SGD could not optimize them. The proposed models and 9-layer deep CNN were trained and validated for 50 epochs. The pre-trained models, that is, MobileNet V2,³ ResNet50 V2,⁴ Inception V3,⁵ and Xception,⁶ were able to achieve comparatively better average accuracy at or before 15 epochs.

9 | EVALUATION METRICS

The use of evaluation metrics for the model's analysis is essential, as they provide the model's limitations. The evaluation metrics used in this study included recognition accuracy, computational complexity, and computation time. Since the model has imbalanced class-wise test samples, its performance was effectively evaluated using the weighted sum of precision, recall, and $F1$ scores. In addition, a detailed analysis of each class was performed using a confusion matrix. On the other hand, the computational complexity of the models was evaluated using floating-point operands (FLOPs), number of parameters, model's size, and computation time using a GPU.

Furthermore, the models' relevance was examined using the Matthew's Correlation Coefficient (MCC), which quantifies the alignment between predictions and actual outcomes. It provides a value between -1 and $+1$. A value in proximity to ± 1 signifies a substantial level of agreement or disagreement, while a value near 0 indicates random prediction. In comparison to Cohen's Kappa and Brier Score, the MCC was preferred because it provides more accurate details.³⁷ Additionally, the comparative significance of the proposed model in relation to top-tier models was evaluated through the application of McNemar's test.³⁸ This approach uses a contingency table, as shown in Figure 5. The elements of the contingency table, that is, a , b , c , and d , represent the total sample count for the corresponding correct or incorrect predictions of the considered models. The mathematical formulation for McNemar's test is given in (2).

$$\chi^2 = \frac{(|b - c| - 1)^2}{b + c}. \quad (2)$$

	model 2 correct	model 2 wrong
model 1 correct	a	b
model 1 wrong	c	d

FIGURE 5 Contingency table for model's significance analysis.

Here, b represents the count of samples correctly predicted by model 1 but incorrectly predicted by model 2, whereas c represents the count of samples correctly predicted by model 2 but incorrectly predicted by model 1. The χ^2 is McNemar's test statistic called chi-squared. The proposed model's significance was tested against the state-of-the-art models with a confidence interval of 90% ($\alpha = 0.10$) and a threshold of 1.303, that is, if $\chi^2 > 1.303$, the proposed model will have a comparatively higher significance.

In addition to these evaluation metrics, we used a statistical approach to collectively evaluate the model's performance while considering average accuracy, FLOPs, model's parameters, and computation time. The mathematical formulation for the approach is discussed as follows:

1. First, the proposed approach normalizes each evaluation metric in a range between 0 and 1 using (3)

$$X_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (3)$$

where, x represents the value we want to normalize, and x_{min} and x_{max} represents the minimum and maximum value of the data under consideration.

2. In the next step, the degree of preference of normalized metrics was adjusted using (4), that is, the average accuracy is preferred high while FLOPs, model's parameters, and computation time

$$X_{pref} = 1 - X_{norm}. \quad (4)$$

3. Finally, the arithmetic mean for each of the calculated metrics was computed using (5), which shows model's overall rank in comparison to other models

$$\text{Mean} = \frac{\text{Average Accuracy} + \text{FLOPs}_{pref} + \text{Parameters}_{pref} + \text{Computation Time}_{pref}}{n} \quad (5)$$

where, FLOPs_{pref} , Parameters_{pref} , and $\text{Computation Time}_{pref}$ are the preferred values of the respective parameters calculated using (3) and (4). Whereas, n represents the number of parameters under consideration, that is, 4 in our case.

Note that the model's size was not considered in (4) as it is directly proportional to the number of parameters.

10 | RESULTS

The proposed and cutting-edge models underwent testing using a PlantVillage dataset comprising 1916 samples, encompassing 39 different classes. The evaluation results of the models for various metrics are listed in Table 1. It was observed that all of the models exhibited a stronger level of relevance, indicated by MCC values approaching +1.

TABLE 1 Proposed models' performance against state-of-the-art models.

Model	Accuracy	Precision	Recall	F1	MCC	Chi-squared
MobileNet V2 ³	0.9942	0.9945	0.9942	0.9941	0.994	37.77
ResNet50 V2 ⁴	0.9441	0.9543	0.9441	0.9461	0.9422	15.22
Inception V3 ⁵	0.9608	0.9728	0.9608	0.9611	0.96	1.54
Xception ⁶	0.9895	0.9899	0.9895	0.9894	0.9891	20.55
9-layer deep CNN ⁷	0.9405	0.9409	0.9405	0.9394	0.9381	27
Proposed model using Standard CNN	0.95929	0.9594	0.9592	0.9585	0.9576	3.9
Proposed model using Separable CNN	0.9566	0.9569	0.9566	0.9558	0.9549	5.83
Proposed model using GCNN	0.9629	0.9631	0.9629	0.9617	0.9614	1.96
Proposed model using ShuffleNet	0.9686	0.9681	0.9686	0.9678	0.9674	-

Furthermore, besides MobileNet V2 and Xception, the proposed model using ShuffleNet has shown greater significance among state-of-the-art models, with $\chi^2 > 1.303$. From the contingency table shown in Figure 6, we observe that the proposed model dominates the comparative models when the “b” is significantly greater than the “c,” and vice versa.

The additional factors pertaining to the model’s performance in Table 1 align with the results obtained from the statistical analysis. Among the proposed models, the model developed using ShuffleNet has dominated by attaining a comparatively higher average accuracy and F1 scores of 96.86% and 96.78%, respectively. The proposed model using ShuffleNet provided overall better results as compared to other state-of-the-art models, except MobileNet V2 and Xception. The F1 scores obtained by state-of-the-art models were 99.41%, 94.61%, 96.11%, 98.94%, and 93.94% for MobileNet V2,³ ResNet50 V2,⁴ Inception V3,⁵ Xception,⁶ and 9-layer deep CNN,⁷ respectively. The confusion matrix of the proposed model using ShuffleNet is shown in Figure 7. Note that the numerical notations from 0 to 38 on the axes correspond to the plant disease classes in Figure 8. The average accuracy for each class was greater than 80%, except for the Tomato Leaf Mold class. The degradation in performance of the respective class is due to the resemblance of the disease symptoms with those of the Tomato Late Blight class. Figure 8 shows the class-wise average accuracy of the proposed model using ShuffleNet as compared to the state-of-the-art models. We observed that most of the models showed a considerable degradation in recognition performance for the Tomato Leaf Mold disease due to the aforementioned cause. In general, the overall class-wise performance of MobileNet V2,³ Xception,⁶ and the proposed model using ShuffleNet was remarkable. MobileNet V2³ and Xception⁶ outperformed the other models but were observed to have high computational complexity. Table 2 provides an overview of the computational complexity of the models, encompassing FLOPs, parameter count, computation time, and model size. We observed that besides the 9-layer deep CNN, the other state-of-the-art models have higher computational complexity as compared to the proposed model using ShuffleNet. Among the state-of-the-art models, ResNet50 V2 and Xception were computationally very expensive, followed by Inception V3 and MobileNet V2. The 9-layer deep CNN model was computationally the least expensive, followed by the proposed model using Separable CNN, GCNN, and ShuffleNet. Despite being computationally least complex, both the 9-layer deep CNN and the proposed model using Separable CNN exhibited lower overall performance in terms of classification accuracy and F1 score compared to

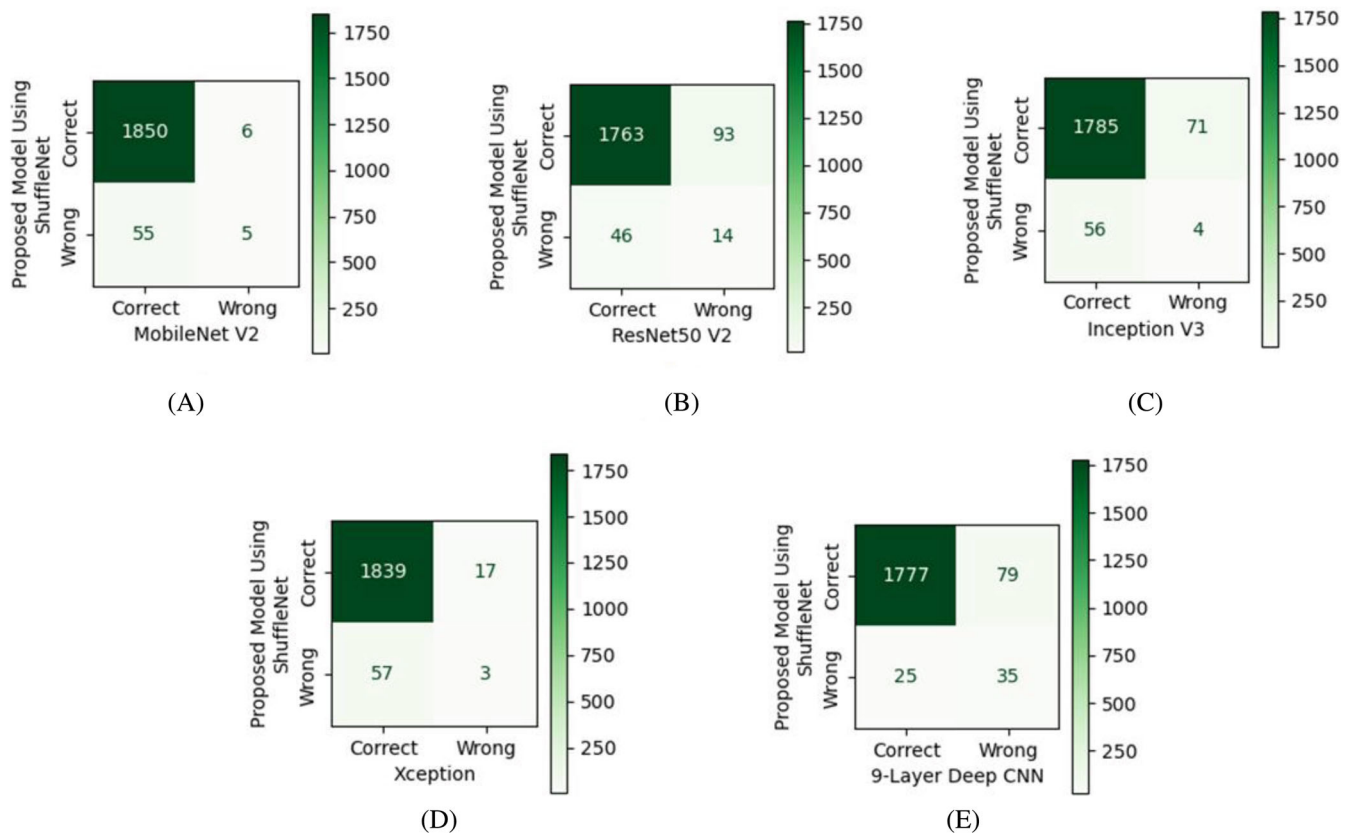


FIGURE 6 Contingency table resulted from the statistical analysis of the proposed model using ShuffleNet, and (A) MobileNet V2, (B) ResNet50 V2, (C) Inception V3, (D) Xception, and (E) 9-layer deep CNN model.

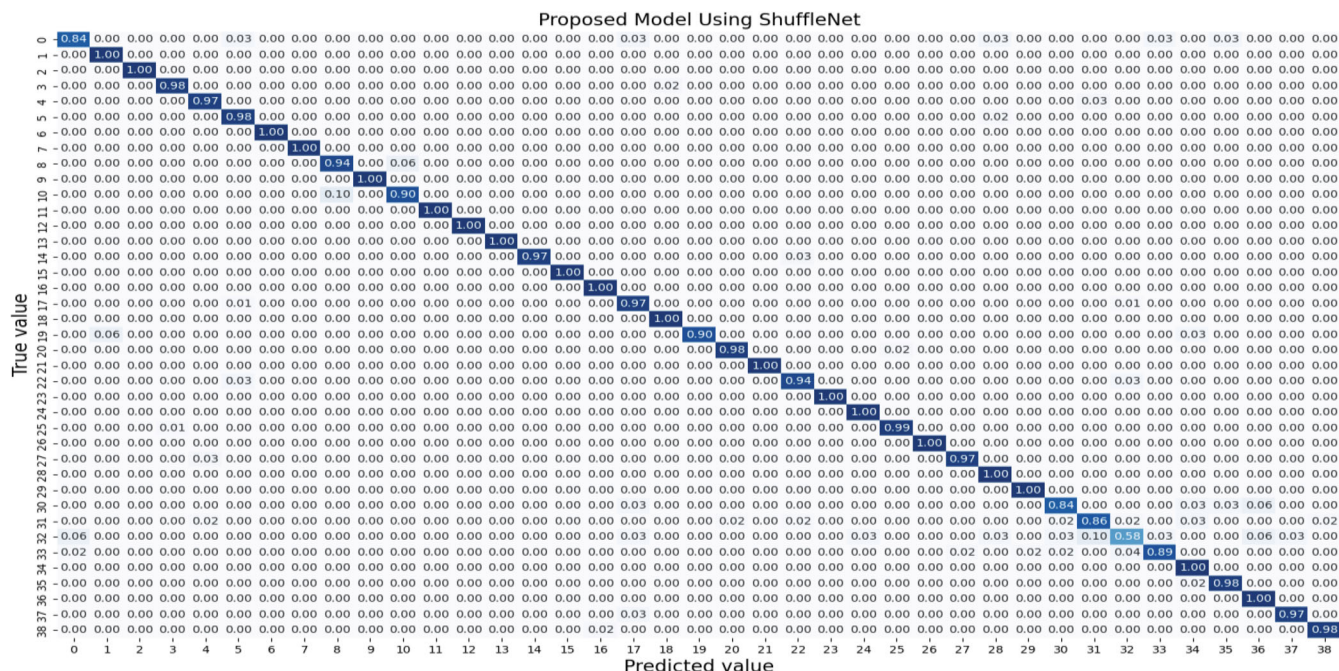


FIGURE 7 Confusion matrix of the proposed model using ShuffleNet.

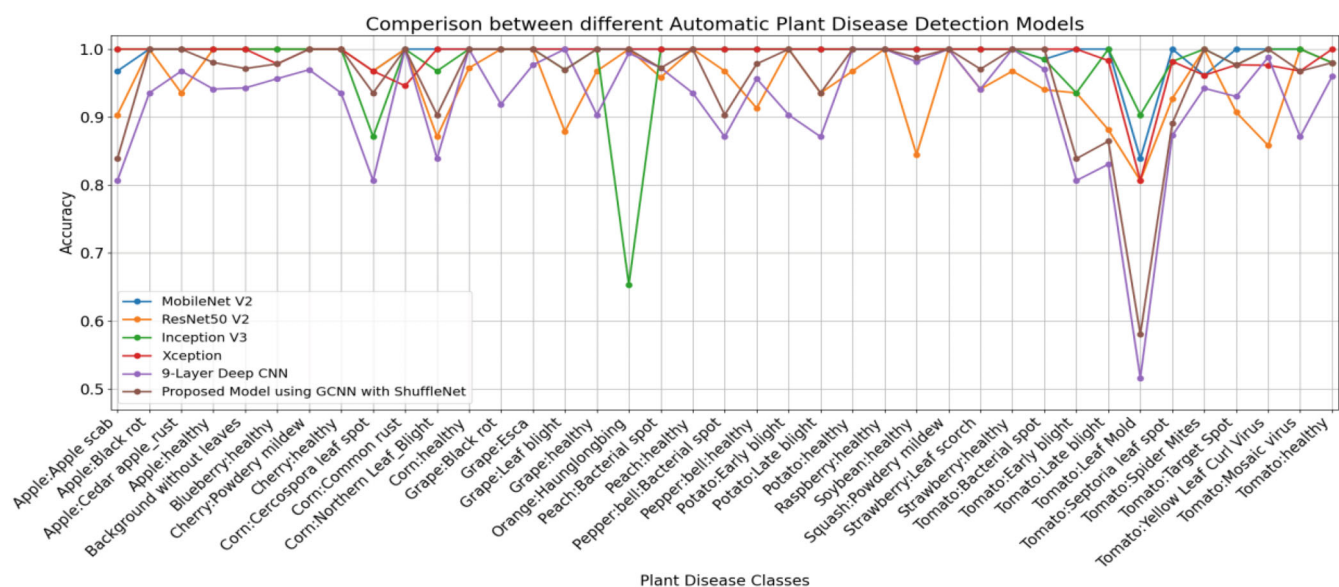


FIGURE 8 Performance analysis of proposed model using ShuffleNet and state-of-the-art models.

the other computationally complex state-of-the-art models. The proposed model using ShuffleNet, on the other hand, showed a fine balance between the model's performance and computational complexity with 37 million FLOPs and 2.1 ms of the computation time. In order to validate the analysis, the collective data analysis was also conducted using (5), as shown in the Figure 9. We have observed that the proposed model using separable CNN outperformed the one built with ShuffleNet. This superiority arises from the former's lower computational complexity, whereas the latter excels in achieving the above-average accuracy. The separable CNN-based proposed model emerges as the preferred option in situations where computational resources are constrained, and it is acceptable to trade off a bit of average accuracy. On the other hand, the ShuffleNet-based model proves ideal when higher average accuracy is required using limited computational resources.

TABLE 2 Computational complexity of proposed models against state-of-the-art models.

MODEL	MFLOPs	Parameters	Computation time (ms)	Model size (MB)
MobileNet V2 ³	124	22,778,023	16.96	87.00
ResNet50 V2 ⁴	1378	56,372,839	85.62	215.00
Inception V3 ⁵	738	30,034,823	59.87	114.00
Xception ⁶	1521	53,669,519	36.98	204.00
9-layer deep CNN ⁷	34	212,727	1.83	0.87
Proposed model using Standard CNN	98	3,309,671	3.36	12.60
Proposed model using Separable CNN	15	606,434	1.98	2.34
Proposed model using GCNN	37	619,111	2.07	2.38
Proposed model using ShuffleNet	37	619,111	2.1	2.39

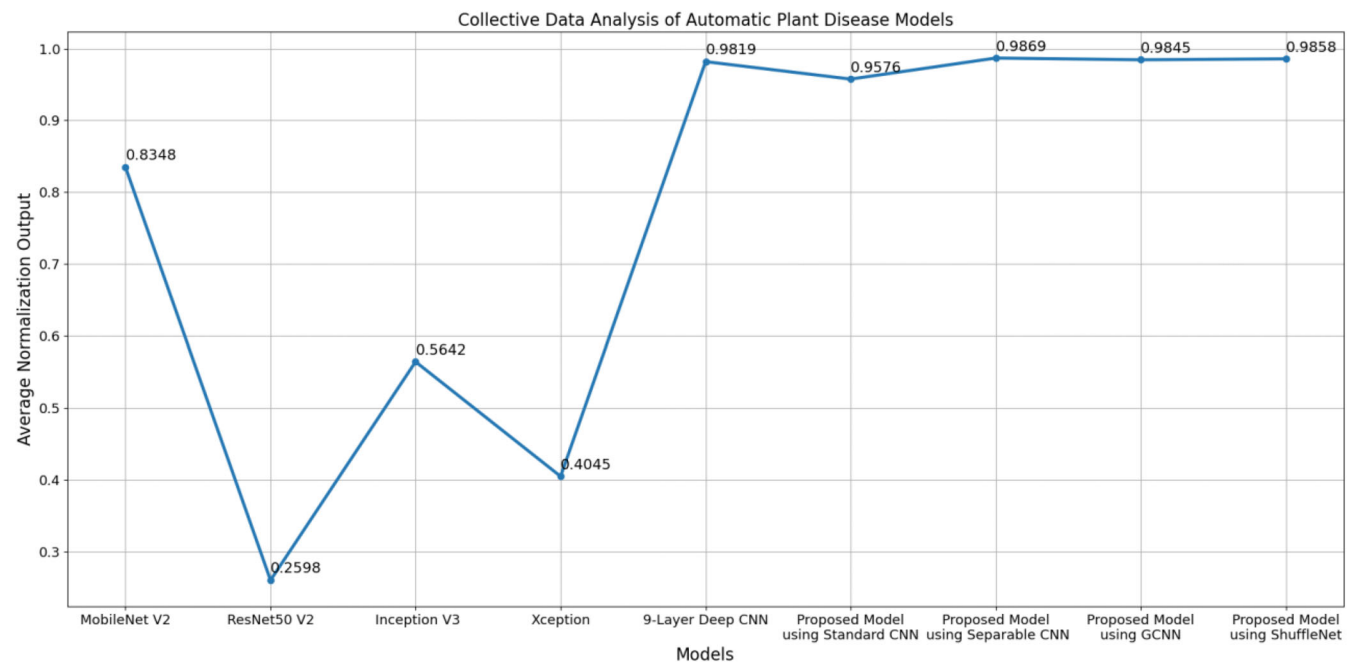


FIGURE 9 Collective data analysis of automatic plant disease models.

The results indicate that, while considering both the model's performance and computational complexity, the proposed model using ShuffleNet dominates the other state-of-the-art models.

11 | DISCUSSION

The automatic plant disease detection approach will be more favorable if it has comparatively higher average accuracy and lower computational complexity. The proposed study evaluated various models and observed that state-of-the-art models, that is, MobileNet V2³ and Xception,⁶ achieved higher average accuracy but were computationally expensive, which limited their usage to a class of farmers. On the other hand, the proposed model developed using separable CNN was observed to have lower computational complexity while compromising the average accuracy. However, the proposed model developed using ShuffleNet was observed to have a balance between average accuracy and computational complexity. The proposed model using ShuffleNet outperformed the state-of-the-art models, including ResNet50 V2,⁴ Inception V3,⁵ and the 9-layer deep CNN,⁷ while having comparatively lower computational complexity. The lower complexity results in a faster response time on resource-constrained devices. Therefore, the proposed model using ShuffleNet provides a promising solution for automatic plant disease detection. The collective data analysis approach provided the best

results for the proposed model using separable CNN due to its low computational complexity and reasonable average accuracy. It was followed by the proposed model using ShuffleNet due to its higher average accuracy while utilizing less computational resources.

12 | CONCLUSION

In the proposed study, a CNN model was developed using a standard convolution approach for automatic plant disease detection. Different CNN variants such as depth-wise separable CNN, GCNN, and ShuffleNet were employed to enhance the proposed model's performance. These variants not only improved the performance of the proposed model but also significantly reduced its computational complexity. The proposed models were compared with state-of-the-art models, that is, MobileNet V2, ResNet50 V2, Inception V3, Xception, and 9-layer deep CNN. In terms of accuracy, MobileNet V2 and Xception provided better results as compared to the proposed models, but these state-of-the-art models were computationally very expensive. The proposed model using ShuffleNet outperformed the state-of-the-art models considering both the accuracy and computational complexity.

The PlantVillage dataset, which was used in this research, involved plucking leaves and placing them on black or gray paper to avoid complex backgrounds. To ensure optimal performance, a similar approach should be adopted when using the proposed model, as its performance may degrade significantly in the presence of complex backgrounds. In addition, the model was trained to detect a 39 plant disease classes. The future goal is to develop a robust model that is capable of automatically detecting plant diseases using leave images with complex backgrounds and a broader range of classes.

AUTHOR CONTRIBUTIONS

Muhammad Rizwan: Conceptualization; investigation; writing – original draft; methodology; software; formal analysis. **Samina Bibi:** Conceptualization; investigation; writing – original draft; methodology; software; formal analysis. **Sana Ul Haq:** Validation; visualization; writing – review and editing; data curation; supervision; resources. **Muhammad Asif:** Validation; visualization; writing – review and editing; data curation; supervision; resources. **Tariqullah Jan:** Visualization; writing – review and editing; validation; data curation. **Mohammad Haseeb Zafar:** Validation; visualization; writing – review and editing; funding acquisition; project administration; data curation.

CONFLICT OF INTEREST STATEMENT

The authors affirm that they do not have any conflicts of interest to disclose regarding the current study.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Mohammad Haseeb Zafar  <https://orcid.org/0000-0001-5637-4798>

REFERENCES

1. Strange RN, Scott PR. Plant disease: a threat to global food security. *Annu Rev Phytopathol*. 2005;43(1):83-116.
2. Saleem MH, Potgieter J, Arif KM. Plant disease detection and classification by deep learning. *Plan Theory*. 2019;8(11):468-490.
3. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. MobileNetV2: inverted residuals and linear bottlenecks. In: *Conference on Computer Vision and Pattern Recognition*, Salt Lake City; 2018:4510-4520.
4. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. In: *European Conference on Computer Vision*, Amsterdam, The Netherlands; 2016:630-645.
5. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. ethinking the inception architecture for computer vision. In: *Computer Vision and Pattern Recognition*, Las Vegas, Nevada; 2016:2818-2826.
6. Chollet F. Xception: deep learning with depthwise separable convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA; 2017:1800-1807.
7. Geetharamani G, Pandian AJ. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput Electric Eng*. 2019;76(1):323-338.
8. Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep neural networks based recognition of plant diseases by leaf image classification. *Comput Intell Neurosci*. 2016;1(1):1-11.

9. Jia Y, Shelhamer E, Donahue J, et al. Caffe: convolutional architecture for fast feature embedding. *arXiv Preprint arXiv: 1408.5093*; 2014.
10. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Commun ACM*. 2017;60(6):84-90.
11. Sardogan M, Tuncer A, Ozen Y. Plant leaf disease detection and classification based on CNN with LVQ algorithm. In: *International Conference on Computer Science and Engineering*, Sarajevo, Bosnia and Herzegovina; 2018:382-385.
12. Hughes DP, Salathe M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv Preprint arXiv: 1511.08060*; 2015.
13. Ferentinos KP. Deep learning models for plant disease detection and diagnosis. *Comput Electron Agric*. 2018;145(1):311-318.
14. Liu S, Deng W. Very deep convolutional neural network based image classification using small training sample size. In: *Asian Conference on Pattern Recognition*, Kuala Lumpur, Malaysia; 2015:730-734.
15. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. OverFeat: integrated recognition, localization and detection using convolutional networks. *arXiv Preprint arXiv: 1312.6229*; 2014.
16. Kamal K, Yin Z, Wu M, Wu Z. Depthwise separable convolution architectures for plant disease classification. *Comput Electron Agric*. 2019;165:1-6.
17. Howard AG, Zhu M, Chen B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv Preprint arXiv: 1704.04861*; 2017.
18. Sagar A, Jacob D. On using transfer learning for plant disease detection. *BioRxiv Preprint*; 2021. doi:10.1101/2020.05.22.110957
19. Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions. In: *Computer Vision and Pattern Recognition*, Boston, Massachusetts; 2015:1-9.
20. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Computer Vision and Pattern Recognition*, Las Vegas, Nevada; 2016:770-778.
21. Szegedy C, Ioffe S, Vanhoucke V, Alemi A. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In: *AAAI Conference on Artificial Intelligence*, San Francisco, California USA. Vol 30, no. 1; 2017:4278-4284.
22. Huang G, Liu Z, Maaten LVD, Kilian Q. Densely connected convolutional networks. In: *Computer Vision and Pattern Recognition*, Honolulu, Hawaii; 2017:4700-4708.
23. Albattah W, Nawaz M, Javed A, Masood M, Albahli S. A novel deep learning method for detection and classification of plant diseases. *Comp Intellig Syst*. 2022;8(1):507-524.
24. Duan K, Bai S, Xie L, Qi H, Huang Q, Tian Q. CenterNet: keypoint triplets for object detection. In: *Computer Vision Foundation*, Seoul, Korea; 2019:6569-6578.
25. Radford A, Kim JW, Hallacy C, et al. Learning transferable visual models from natural language supervision. *arXiv Preprint arXiv:2103.00020*.
26. Liu C, Cheng S, Chen C, et al. M-FLAG: medical vision-language pre-training with frozen language models and latent space geometry optimization. *arXiv Preprint arXiv:2307.08347*.
27. Wan Z, Liu C, Zhang M, et al. Med-UniC: unifying cross lingual medical vision-language pre-training by diminishing bias. *arXiv Preprint arXiv:2305.19894*.
28. Fenu G, Mallocci FM. DiaMos plant: a dataset for diagnosis and monitoring plant disease. *Agronomy*. 2021;11(11):1-12.
29. Singh D, Jain N, Jain P, Kayal P, Kumawat S, Batra N. PlantDoc: a dataset for visual plant disease detection. In: *CoDS COMAD 2020: 7th ACM IKDD CoDS and 25th COMAD*, Hyderabad, India; 2020:249-253.
30. Liu X, Min W, Mei S, Wang L, Jiang S. Plant disease recognition: a large-scale benchmark dataset and a visual region and loss reweighting approach. *IEEE Trans Image Process*. 2021;30:2003-2015.
31. Bonnici E, Arn P. *The Impact of Data Augmentation on Classification Accuracy and Training Time in Handwritten Character Recognition*. KTH Royal Institute of Technology; 2021.
32. Aquino MR, Gutoski M, Hattori L, Lopes HS. The effect of data augmentation on the performance of convolutional neural networks. In: *Brazilian Congress of Computational Intelligence*, Niterói, Rio de Janeiro, Brazil; 2017.
33. Zhang X, Zhou X, Lin M, Sun J. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA; 2018:6848-6856.
34. Laha SR, Samal S, Nayak DSK, Pattanaik S, Pattanayak BK. Challenges and solution for identification of plant disease using machine learning and IOT. *Int J Comput Commun Technol India*. 2023;9(1):40-44.
35. Kim P. *Overfitting*. *Matlab Deep Learning*. 1st ed. Apress; 2017:6.
36. Kingma DP, Ba J. Adam: a method for stochastic optimization. *arXiv Preprint arXiv: 1412.6980*; 2014.
37. Chicco D, Warrens MJ, Jurman G. The Matthews correlation coefficient (MCC) is more informative than Cohen's kappa and brier score in binary classification assessment. *IEEE Access*. 2021;9(1):78368-78381.
38. Edwards AL. Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. *Psychometrika*. 1948;13(1):185-187.

How to cite this article: Rizwan M, Bibi S, Haq SU, Asif M, Jan T, Zafar MH. Automatic plant disease detection using computationally efficient convolutional neural network. *Engineering Reports*. 2024;6(12):e12944. doi:10.1002/eng2.12944