International Journal of Basic and Applied Sciences, 14 (7) (2025) 595-605

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/myqfsc31
Research paper

Fast and Efficient CNN Architectures for Automated Soybean Leaf Disease Classification

Kalpesh Patel [1] *, Atul Patel [2]

[1] *College of Agriculture, Anand Agricultural University, Vaso, Kheda, Gujarat, India*
[2] *Smt. C M Patel Institute of Computer Applications, Charotar University of Science and Technology, Changa, Anand, Gujarat, India.*
**Corresponding author E-mail: kpatel@aau.in*

Received: September 5, 2025, Accepted: November 20, 2025, Published: November 29, 2025

Abstract

Soybean leaf diseases significantly affect crop yields in Indian agriculture. Timely detection is crucial for managing these diseases effec-tively-ly. While traditional CNN models deliver high accuracy, they have significant computational demands and slower speeds, limiting real-time use. This study aims to create faster CNN models that effectively identify eight common soybean leaf diseases while being easier to train. The method uses a Separable-CNN with a Global Average Pooling (GAP) layer to reduce parameters and computing needs. The dataset used to train the Fast CNN and Enhanced CNN models included 40,000 images across eight disease categories. These models underwent evaluation using validation and test datasets with varying splits: 70-20-10, 65-25-10, and 60-30-10. The Enhanced CNN achieved 98.14% test accuracy, outperforming models like VGG16, ResNet152V2, and InceptionV3. The Fast CNN achieved 95.70% accuracy with reduced training time, making it suitable for real-time deployment. The confusion matrix showed high accuracy with few errors. The new CNN models accurately identify soybean leaf diseases and work efficiently. These models could help manage diseases quickly and improve AI tools in agriculture.

Keywords: *Deep Learning; Separable CNN; Global Average Pooling; Soybean Leaf Disease; Computational Complexity.*

1. Introduction

In India, soybean represents a principal oilseed crop, with its production being considerably impacted by various foliar diseases. The reliance on visual inspection by farmers or agricultural experts is susceptible to human error and may fail to identify diseases at their incipient stages. Rapid and accurate identification of these diseases is essential for prompt intervention and successful management. As a result, the need for more effective and precise disease detection techniques has driven the creation of cutting-edge technologies in the field of precision agriculture.

New technologies like image processing and machine learning are changing how soybean leaf diseases are found in Indian farming. They make the process faster and more accurate. Computer vision and machine learning techniques have transformed leaf disease detection across various plant species, enabling automated and accurate identification of subtle disease signs[1]. Recent progress has created ad-vanced image processing methods and deep learning models. These use Convolutional Neural Networks to detect diseases accurately [2]. These tools can greatly improve soybean farming in India. They help find diseases early, allow for specific treatments, and improve crop yield and quality. Artificial intelligence and deep learning present promising avenues, especially through the use of Convolutional Neural Networks (CNNs). Nonetheless, conventional CNN models are hindered by significant computational demands and risk of overfitting, rendering them impractical for Indian farms with limited resources.

This study introduces an innovative enhanced model of Separable Convolutional Neural Networks (Separable-CNNs) that incorporates a Global Average Pooling (GAP) structure. The architecture utilizes depth-wise separable convolutions to decompose traditional convolution into depth-wise and point-wise operations, thereby lowering the computational complexity while preserving the accuracy. A GAP layer enhances the model by converting the feature maps into a comprehensive global feature, which boosts the robustness of the extraction and minimizes overfitting.

The primary aim of this research was to develop a robust model for the identification of soybean leaf diseases across various agricultural settings in India. This study was guided by three principal objectives:

- Model Design and Implementation: To create and deploy a Separable-CNN model that integrates a Global Average Pooling layer for detecting diseases in soybean leaves.
- Performance Assessment: To assess the model's effectiveness using a dataset of soybean leaf images, concentrating on metrics such as precision, recall, and accuracy.

- Real-World Application: To evaluate the model's feasibility for real-time disease detection in agricultural fields, ensuring its practical utility for farmers.

This study proposed that the detection of soybean leaf diseases using an enhanced Separable-CNN integrated with a GAP layer is in line with the latest progress in plant disease detection via deep learning methods. This method addresses the demand for effective and precise disease detection techniques in agriculture, particularly for soybean cultivation in India. Separable-CNN, utilizing depth-wise separable convolutions, presents a promising approach to optimizing computational resources while ensuring high accuracy.

Separable-CNNs with Global Average Pooling offer a novel method for boosting model efficiency and performance. Separable convolutions, especially depth-wise separable convolutions, have demonstrated their ability to decrease the number of parameters and computational complexity while preserving accuracy [3], [4]. This approach involves dividing the standard convolution into a depthwise convolution followed by a pointwise convolution, which significantly reduces model size and enhances inference speed.

Global Average Pooling (GAP) helps models work more efficiently and effectively. It replaces the usual fully connected layers at the network's end, which reduces the number of parameters and helps prevent overfitting [3], [5]. GAP has shown better results in tasks like image and audio recognition [6]. Some researchers have also looked into other pooling methods, like Deep Generalized Max Pooling [7], which might be useful in certain situations. The combination of Separable-CNNs and GAP is a promising way to create better and more efficient neural networks. This method helps reduce model size and complexity while keeping or improving performance on different tasks. Future studies could look into adding attention mechanisms [8] or new pooling methods to make these models even better.

The Separable-CNN incorporating a GAP layer for identifying soybean leaf diseases in India appears to be a promising lightweight deep learning model for plant disease detection. This method, which is computationally efficient, has the potential to facilitate real-time agricultural applications, thereby enhancing crop management and promoting sustainability.

2. Literature Review

Recent studies on detecting diseases in crop leaves use techniques like image processing, machine learning, and deep learning. These methods are very accurate [9]. They have mostly replaced old manual checks, making crop analysis more accurate and cheaper.

The integration of hyperspectral imaging with UAV technology has shown great promise in the field of plant disease detection. UAV-mounted HRS systems deliver superior spectral, spatial, and temporal resolutions for crop analysis.[10]. By applying a CNN-based VGG16 model, researchers attained a training accuracy of 95.16% and a testing accuracy of 93% in detecting diseases in maize leaves [11]. In a similar vein, a new deep mutual learning model named DVNet, which utilizes the Densenet 121 and VGG19 neural networks, successfully achieved an accuracy rate of 94.72% in identifying eight different disease categories in mango leaves [12].

Convolutional Neural Networks (CNNs) are effective tools for identifying plant diseases. They outperform traditional methods. CNNs are used to detect diseases in various crops like tomatoes, cucumbers, rice, and soybeans. For rice, using CNNs with enhanced data techniques has achieved a high accuracy of 99.7%.[13]. In the realm of soybean disease detection, transfer learning techniques utilizing pretrained AlexNet and GoogleNet CNNs have achieved impressive accuracy rates of 98.75% and 96.25%, respectively [14]. While CNNs have shown exceptional results, some research has investigated other methods. For instance, a capsule neural network (CapsNet) was introduced for detecting tomato leaf diseases, reaching an accuracy of 96.39% and exhibiting a better capability to capture spatial positioning in images compared to conventional CNNs.[15]. Recent studies have looked into using transformer networks to detect plant diseases. They achieved a validation accuracy of 97.98% [16]. CNNs are successful in this area because they can automatically find important features in images. This reduces the need for manual work to identify these features.

CNN-based techniques for detecting leaf diseases have demonstrated notable success, yet they encounter several obstacles. A significant challenge is their inability to effectively capture the spatial and orientation relationships within leaf images, which are essential for accurately identifying disease spots based on their shape, color, and location [15]. This limitation can result in decreased precision when identifying and categorizing certain plant diseases. CNN models typically demand extensive training datasets and significant computational power, posing challenges for deployment on embedded mobile devices with restricted capabilities [17]. This restriction impedes the broad application of these methods in practical agricultural environments where quick, on-site disease detection is crucial. Moreover, CNN models often face the challenge of overfitting, especially when working with limited or unbalanced datasets [18], [19]. This issue can lead to inadequate generalization and lower accuracy when the models are used on new, unseen leaf images. In addition, the requirement for extensive preprocessing and feature extraction techniques can add to the complexity of implementing CNN-based methods [20].

Convolutional Neural Networks (CNNs) have experienced substantial progress in recent years, with researchers delving into a variety of innovative strategies to boost their performance and efficiency. A prominent trend is the utilization of spatial and channel data, alongside the investigation of depth, width, and multipath information processing. Moreover, the adoption of layer blocks as structural units has become increasingly prevalent in the architecture of CNNs. [21]. Architectures like AlexNet, VGGNet, and GoogleNet have highlighted the influence of altering network depth and width on their effectiveness. Notably, VGGNet, with its extensive design comprising 13 convolutional layers, has proven to be particularly effective in tasks such as identifying plant diseases.[22]. In recent developments, preprocessing techniques such as Spatial Rich Models filters, Spatial Dropout, Absolute Value layer, and Batch Normalization have been employed to enhance the performance, accuracy, and stability of CNNs during steganalysis training[23]. By extending and deepening network architectures and incorporating cutting-edge preprocessing methods and feature selectors, these innovations have significantly expanded the capabilities of CNNs across a wide range of applications.

As shown by current studies, deep CNNs and their extensions are capable of high accuracy in detecting plant diseases in a variety of crops [9]-[22]. Nonetheless, there are still a number of gaps. To begin with, several of them are based on heavy architectures, including deep VGG, ResNet, or Inception variants, which are highly accurate, but very memory, computationally, and hardware-intensive, thus not easily able to be deployed to resource-constrained platforms, which are commonly found in agricultural environments [17], [24]. Second, despite new research suggesting lightweight or compressed plant disease recognition models [24]-[26], many of them are restricted to a small set of disease categories or crops, not soybean, and many of them are not specifically looking at field-realistic data that is measured in an Indian agronomic setting. Third, others mainly measure performance using pre-existing benchmark data sets and fail to present a specific analysis of the trade-off between accuracy, model size, and training/ inference time, which is important in real-time decision support tools in precision agriculture.

The current research fills these gaps by developing and assessing three CNN networks to classify soybean leaf disease automatically. The CNN base delivers a traditional high-capacity reference model. Based on this, the Fast CNN uses separable convolutions, Global Average Pooling, and aggressively pruned down parameter count to provide competitive accuracy in a significantly smaller model with a significantly shorter training time, suitable for real-time use on relatively small hardware. The Enhanced CNN uses depthwise and conventional convolutions with GAP and dropout to co-optimize accuracy and efficiency, which shows better results on test (up to 98.14% accuracy) and is significantly lighter than most typical transfer-learning models such as VGG16, ResNet152V2, and InceptionV3. With an eight-class dataset of soybean diseases gathered in the Indian fields and explicitly evaluating the trade-offs between the accuracy, the model complexity, and training time, this work provides a practically deployable and empirically validated solution to AI-based detection of diseases in soybean farming.

3. Methodology

3.1. Dataset

To ensure the authenticity and applicability of the dataset to real-world conditions, the data on soybean leaf deficiencies used in this study were carefully selected after genuine field photographs were accurately captured, as illustrated in Figure 1. The curation process involves several key stages. To understand soybean leaf diseases better, images have been taken directly in the fields of the soybeans, attempting to mimic the conditions of natural farming. The data was recorded in various areas related to the University and the adjacent areas, at different stages of the growth of the crops, with the natural variations in the leaf orientation, severities of the diseases, as well as clutter of the backgrounds. The original field photographs were initially annotated with agricultural experts and manually blacked in the background, to remove non-informative background (soil, weeds, and sky), to increase label reliability, and to improve the photos to be more suitable for deep learning. The size of every image was reduced to 224 x 224 pixels so that they can fit in the typical deep learning structures. The obtained dataset is 40,000 images of soybean leaves represented in eight classes (seven disease types and one healthy type). The images cover a variety of illumination conditions (e.g., overcast, partly cloudy, mild shadows) and were collected at different growing periods, which enhances the soundness of the learned representations and facilitates the usage of the models in a variety of field conditions.

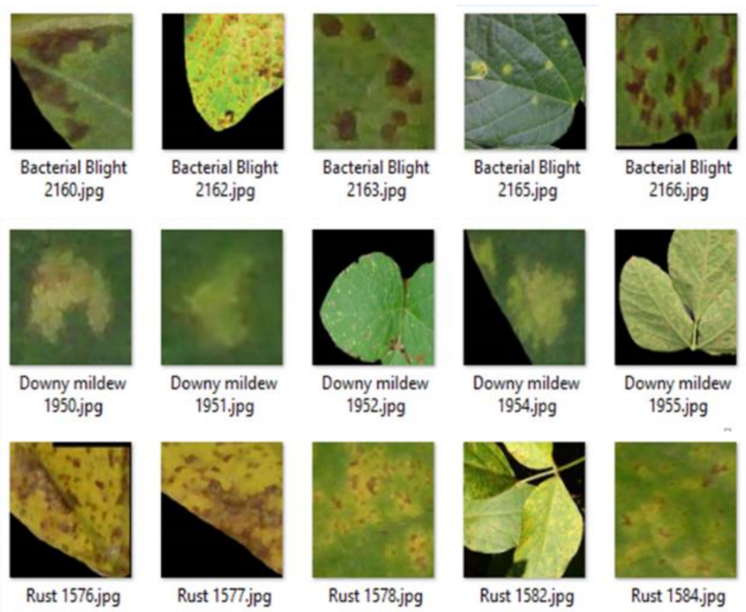

Fig. 1: Soyabean Leaf Dataset Images.

Table 1: Dataset Images Counts

No	Disease Name	Number of Images
1	Mosaic-virus	5000
2	Powdery mildew	5000
3	Bacterial-Blight	5000
4	Downy mildew	5000
5	Rust	5000
6	Southern blight	5000
7	Septoria brown spot	5000
8	Healthy Soybean	5000
	Total	40000

3.2. Transfer learning models

Recent developments in image processing have been greatly shaped by the advent of deep convolutional neural networks (CNNs), especially those designs that effectively balance computational efficiency with the ability to represent complex data. Among the foundational deep architectures are VGG-16 and VGG-19, introduced by Simonyan and Zisserman [27], which are notable for their emphasis on depth and simplicity. These architectures utilize a uniform arrangement of small 3×3 convolutional kernels throughout the network. VGG-16 consists of 13 convolutional layers and 3 fully connected layers, whereas VGG-19 includes 16 convolutional layers. Despite their high computational cost and parameter count, both networks are widely used for transfer learning in image classification, segmentation, and enhancement tasks owing to their consistent feature representation.

ResNet152V2 is an enhanced version of the original Residual Network, utilizing identity mappings and pre-activation residual blocks to tackle the vanishing gradient issue, which allows for the successful training of deeper networks. This design is especially advantageous for intricate tasks like texture analysis and detailed object recognition. Incorporating identity mappings as skip connections enables the direct transmission of forward and backward signals between blocks, making training more straightforward and enhancing generalization [28].

The pre-activation design, which places batch normalization and activation before convolutions, further enhances the performance of the network. In addition, the development of capsule networks with residual pose routing has demonstrated effectiveness in image classification tasks, offering a different perspective on deep architecture design [29].
The Xception architecture, created by Chollet, is a big step forward in designing convolutional neural networks. It uses depthwise separable convolutions [30]. This method splits spatial filtering and feature mixing across channels. This makes it less complex to compute but still very accurate for classifying images. Depthwise separable convolutions are very efficient in many uses. They work well in top networks like MobileNets and Xception, showing they are good at balancing performance and resource use [31]. The XceptionTime framework, designed for the recognition of sparse multichannel surface Electromyography (sEMG) signals, enhances the Xception model by integrating depthwise separable convolutions, adaptive average pooling, and an innovative non-linear normalization method [32].
Inception_v3, created by Szegedy et al. [33], improves upon earlier Inception models by integrating factorized convolutions, asymmetric filters, and auxiliary classifiers to enhance both precision and performance. The Inception framework has demonstrated exceptional capability in capturing multi-scale features across a range of image-processing tasks. The Inception Recurrent Convolutional Neural Network (IRCNN) merges the strengths of Inception networks with recurrent convolutional layers, resulting in superior performance in object recognition tasks [34]. This architecture extends the capabilities of both Inception and RCNN models, achieving higher recognition accuracy than well-known DCNNs on benchmark datasets like MNIST, CIFAR-10, CIFAR-100, and SVHN.
Inception-ResNetV2 is a neural network that combines Inception modules with residual connections, making it more efficient and faster to train. It is effective for many computer vision tasks [35]. By blending Inception and residual structures, it can gather features in multiple ways while keeping the benefits of identity mappings. This results in quicker training and improved accuracy. These advanced designs set new standards on large datasets like ImageNet and are also effective for other image tasks like noise removal, resolution enhancement, object detection, and medical image analysis.

3.3. Proposed CNN architecture

This research introduced three models based on Convolutional Neural Networks (CNN) for classifying images of soybean crops: a conventional CNN model, a Fast CNN model, and an Enhanced CNN model. Each model was crafted with distinct architectural and hyperparameter settings to assess its effectiveness and efficiency in processing soybean crop image data.
The standard CNN model includes a Leaky ReLU and convolution layer kernels. We selected the CNN architecture for the transfer learning models based on the requirements of our study. By designing a CNN from scratch, we could customize the layers, kernel sizes, and activation functions to optimize the performance for soybean leaf disease classification, providing greater flexibility and a targeted solution aligned with our objectives.

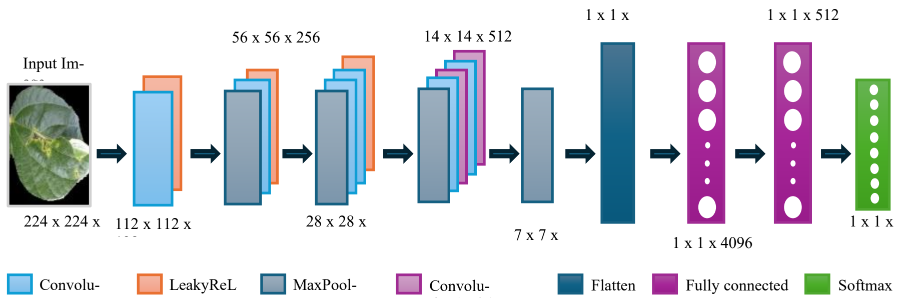

Fig. 2: Standard CNN Model Architecture.

Figure 2 shows the CNN architecture of the soybean leaf disease classification model. The first layer uses a 7 × 11 kernel size in Conv2D, followed by Leaky ReLU in the second layer. The third layer is for Max Polling, which is repeated three times. The last layer has dense and SoftMax to indicate the disease class. The kernel in con2d divides the image into 7 × 11 blocks to gather pixel information.
Exploring extensive convolutional Inception, ResNets, VGGs, and other CNN models prompts an inquiry into how to decrease the parameter count in these networks while maintaining or even enhancing their accuracy and generalizability. This is demonstrated in Figure 3 of the proposed Fast CNN model architecture, which deconstructs the networks into layers. The answer lies in a separable CNN layer with GAP. The Fast CNN model was optimized for computational efficiency and speed. It consists of five separable convolutional layers, which reduce the number of parameters and computational load by factorizing standard convolutions. This architecture also includes five BatchNormalization layers for stabilizing and accelerating the training process and five Max Pooling layers for downsampling. Additionally, two Global Average Pooling layers were integrated to further reduce dimensionality before the final classification stage, as shown in Table 2.

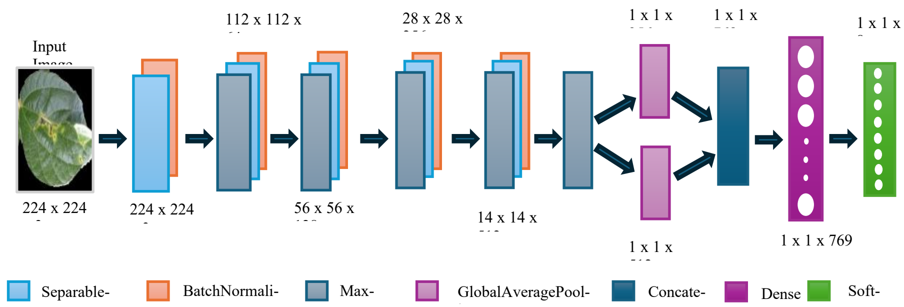

Fig. 3: Fast CNN Model Architecture.

Table 2: Fast CNN Model Hyper Parameters

Crops	soybean
Separable Convolutional Layers	5
BatchNormalization Layers	5
Max Pooling Layers	5
Global average pooling Layers	2
Activation Function	ReLU, Softmax
Epochs	100
Optimizer	Adam (Default lr = 0.001)
Learning Rate	0.0001
Image Size	224 x 224 x 3
Batch Size	32

The Enhanced CNN model in Figure 4 combines the strengths of both standard and fast architectures by incorporating five depthwise convolutional layers alongside five standard convolutional layers to extract both channel-wise and spatial features efficiently. The network further includes five Batch Normalization layers, five Max Pooling layers, and two Global Average Pooling layers, along with a dropout rate to enhance the generalization. The activation functions used were ReLU and Softmax, similar to the Fast CNN. This model also used the Adam optimizer with a learning rate of 0.0001 and was trained over 100 epochs with a batch size of 32, as shown in Table 3.

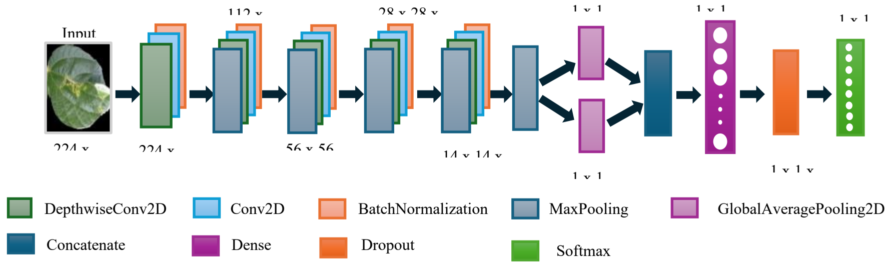

Fig. 4: Enhanced CNN Model Architecture.

Table 3: Enhanced CNN Model Hyper Parameters

Crops	soybean
Depthwise Convolutional Layers	5
Convolutional Layers	5
BatchNormalization Layers	5
Max Pooling Layers	5
Global average pooling Layers	2
Dropout Rate	0.50
Activation Function	ReLU, Softmax
Epochs	100
Optimizer	Adam (Default lr = 0.001)
Learning Rate	0.0001
Image Size	224 x 224 x 3
Batch Size	32

All three models were tailored to the soybean crop classification task and evaluated based on their ability to balance the accuracy, training efficiency, and computational complexity.

3.4. Separable convolution

Spatially Separable Convolutions
Several sub-processes were utilized to achieve the same outcome as the main process. Separable Convolutions are primarily divided into two categories: spatially separable convolutions and those that incorporate deep-learning techniques. The dimensions of the images were characterized by their height and width. A spatially separable kernel facilitates the separation along these spatial dimensions. By splitting the kernel into two segments and applying each segment to the input image in succession, the same result was obtained using the complete kernel.
To improve computational efficiency and reduce the number of trainable parameters in the model, the proposed Fast CNN model employs spatially separable convolutions. This technique decomposes a standard 2D convolution into two separate one-dimensional operations: a horizontal (1 × k) convolution followed by a vertical (k × 1) convolution. For example, a standard 2D convolution with a 3 × 3 kernel requires nine multiplications per filter per pixel. In contrast, a spatially separable convolution first applies a 3 × 1 convolution (three multiplications), and then a 1 × 3 convolution (three multiplications), totaling only six multiplications per filter per pixel, a reduction of 33%. The results are shown in Figure 5.

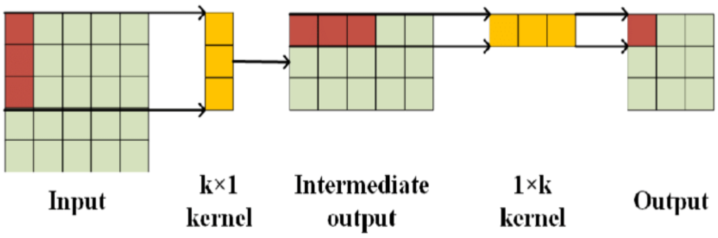

Fig. 5: Spatially Separable Convolution.

This factorization not only decreases computational complexity but also minimizes the number of parameters, making the model more lightweight and faster to train. Despite this simplification, spatially separable convolutions retain the ability to capture spatial dependencies in both horizontal and vertical directions, thus maintaining strong performance in visual tasks such as soybean crop classification. This makes them particularly suitable for real-time agricultural applications, where processing speed and memory constraints are critical.

3.5. Global average pooling (GAP)

To enhance model efficiency and minimize the risk of overfitting, Global Average Pooling (GAP) layers were incorporated into the newly developed Fast CNN and Enhanced CNN models. Unlike traditional fully connected (dense) layers, which significantly increase the number of parameters, GAP operates by computing the average of each feature map, effectively reducing each channel to a single value. This results in a drastic reduction in the number of parameters, while preserving the spatial and semantic information learned by the convolutional layers. For example, if the final convolutional layer outputs a feature map of size 7 × 7 × 256, a fully connected layer flattens this into a 12,544-dimensional vector, leading to a high number of trainable parameters. Conversely, a Global Average Pooling (GAP) layer reduced this to a 1 × 1 × 256 vector by averaging each 7 × 7 feature map, which was subsequently transmitted directly to the classification layer. This approach not only minimizes memory usage and computational demands but also serves as a structural regularizer, promoting the network's ability to discern the most globally pertinent features. In the context of soybean crop classification, GAP enables models to sustain high accuracy while markedly reducing complexity, rendering them ideal for real-time applications in agriculture, where computational resources may be constrained.

To sum up, incorporating GAP layers into CNN frameworks has been widely recognized as an effective strategy for enhancing model efficiency and minimizing overfitting. This method has been effectively utilized across multiple fields, such as fault diagnosis, plant disease detection, and human activity recognition [24], [36].

3.6. Depthwise separable convolutions and pointwise convolution

Depthwise separable convolutions represent a specialized form of separable convolution that breaks down the conventional convolution operation into two sequential stages: depthwise convolution and pointwise convolution. This technique substantially lowers both the computational demands and the parameter count, while still maintaining the performance levels of deep learning models [37], [38]. In the depthwise convolution stage, each input channel is processed with a unique filter, and the pointwise convolution stage utilizes a 1 × 1 convolution to integrate the outputs from the depthwise stage. This division results in a notable decrease in parameters and computational requirements compared to traditional convolutions [39], [40]. In tasks involving 3D vision, depthwise separable convolutions can significantly cut down the number of parameters needed [40]. These convolutions are like a more advanced form of the Inception module, featuring many towers [39]. This concept led to the creation of architectures like Xception, which surpasses Inception V3 in handling large image classification tasks. The efficiency of depthwise separable convolutions makes them ideal for lightweight models used in mobile and embedded devices, such as MobileNets [38], [41].

The Enhanced CNN model integrates depthwise separable convolutions with pointwise convolutions to form an architecture that is both efficient and lightweight, maintaining accuracy. This method decomposes a standard convolution into two distinct operations: depthwise convolution, which applies a single filter to each input channel for spatial filtering, and pointwise convolution, which uses a 1 × 1 convolution to integrate the outputs from the depthwise operation, enabling channel mixing. This separation substantially lowers the computational complexity and reduces the number of parameters compared to traditional convolution methods. For instance, a standard 3 × 3 convolution applied to a 32-channel input with 64 filters requires 3 × 3 × 32 × 64 = 18,432 multiplications. Conversely, depthwise separable convolution requires 288 operations for the depthwise phase, calculated as 3 × 3 × 32, and 2,048 operations for the pointwise phase, calculated as 1 × 1 × 32 × 64. This results in a total of only 2,336 multiplications, representing nearly an eightfold decrease in computational effort.

This architectural refinement allows the Enhanced CNN model to effectively extract both spatial and channel-wise features, enabling faster training and inference while maintaining high classification accuracy. In the context of soybean crop analysis, the use of depthwise separable convolutions and pointwise convolutions ensures that the model is both accurate and scalable, making it suitable for real-time agricultural decision-making systems and deployment in low-resource environments.

3.7. Data augmentation techniques

To increase the robustness and generalization ability of the proposed models and minimize overfitting, different data augmentation methods were used in the training stage. Random geometric transformations (horizontal and vertical flip, small rotation, width and height shift, and moderate zoom) were used on each minibatch. These changes are simulations of typical variations in field imagery in terms of differences in camera angle, leaf orientation, and distance to the canopy. Further, there were small perturbations of intensity and contrast to simulate illumination changes. With such network transformations revealed with each training, network size and diversity are enhanced without necessitating further manual labelling. This promotes the models to acquire disease-specific patterns that are invariant to pose changes, scale changes, and mild lighting changes, thus minimizing chances of memorizing training samples and enhancing the ability to generalise to previously unseen images.

3.8. Training process and hyperparameter optimization

The training setup for the CNN models was designed to help them learn and perform well. Each model went through 100 training cycles with a batch size of 32, which balanced speed and stability. The Adam optimizer was chosen because it adjusts well and handles sparse data effectively. Although Adam usually uses a learning rate of 0.001, a lower rate of 0.0001 was used instead. This smaller rate allowed for more precise updates and helped avoid errors during training, leading to more stable results over time.

Hyperparameter optimization was conducted through a process of iterative experimentation, where various architectural setups and parameter values were evaluated to determine the most effective configuration for classifying soybean crop images. Essential hyperparameters, including the number of convolutional layers, activation functions (such as ReLU, LeakyReLU, and Softmax), dropout rates, and pooling methods, were adjusted based on validation results. Furthermore, techniques like Batch Normalization and Global Average Pooling were incorporated into certain model variants to enhance convergence speed and minimize overfitting. The strategic selection of training parameters and hyperparameter tuning was crucial in improving the accuracy, efficiency, and generalization abilities of the proposed models.

We also used joint architectural and training-level regularization to further regularize the model. To stabilize activation distributions and ensure faster convergence, Batch Normalization layers were added to every Fast and Enhanced CNN model, and the Global Average Pooling layers at the end of the convolutional backbone are a replacement for high-dimensional fully connected layers, which reduces the number of trainable parameters by a significant amount and serves as a structural regularizer. The Enhanced CNN had a dropout rate of 0.50 before the final classification layer, which randomly deactivates neurons during training, and which the authors claim will not encourage co-adaptation of features. The selection of the model was done on the basis of the optimal validation performance, and the three suggested models were tested on several data splits (70-20-10, 65-25-10, and 60-30-10) to determine the robustness. The consistency in these splits (Table 5) suggests that the models are generalizable and not highly optimized to one of the splits of the data.

3.9. Evaluation metrics and performance assessment

To comprehensively evaluate the performance of the proposed CNN-based models in soybean crop classification, a range of standard evaluation metrics was employed. These metrics offer a detailed understanding of the model's effectiveness across various classification dimensions. The evaluation of CNN-based models for tasks like soybean crop classification relies on these metrics to provide a thorough perspective on model performance. Commonly used metrics include accuracy, precision, recall, and F1-score, all derived from the confusion matrix components [42 - 45]. The primary metrics used were accuracy, precision, recall, and F1-score, each calculated using the confusion matrix elements: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).
The classification performance of each model was assessed by calculating these metrics for both the training and validation datasets. By incorporating a range of metrics, the proposed system provides a thorough evaluation of model behavior, particularly in scenarios with class imbalance or varied prediction difficulties. This method facilitates the effective selection and improvement of models, ensuring their reliable application in real-world agricultural contexts.

4. Results and Discussion

In this analysis, we conducted a comparison between baseline models and standard machine-learning methods. These comparisons are essential for situating the proposed models for plant disease classification within a broader context and for assessing their efficacy. To bridge this gap, the following measures were implemented: incorporating widely recognized baseline models in plant disease classification could enhance the article, making it a valuable resource. Examples of traditional machine-learning models include Xception, InceptionV3, VGG16, ResNet152V2, InceptionResNetV2, and InceptionV3. By integrating these models, we can assess the performance of Separable-CNN with global average pooling in comparison to other established techniques. Herein, we present the findings of various models and a comparative analysis that considers multiple factors.
The original photographs were obtained in the field conditions, but the models were trained on pictures whose backgrounds were blackened. Consequently, the trained convolutional filters are subjected, in training, to natural variability in the manifestation of diseases, leaf pose, and illumination, where the background removal step minimizes noise in non-plant areas. In practice, however, leaves of soybean can be partially covered by other leaves or steam or farm equipment, and the lighting can be more extreme (e.g., heavy shadows, backlight, or low light at dusk). The extensive data augmentation entails the introduction of rotations, translations, zoom, and a bit of geometric distortions that, to some extent, mimic these effects, and Global Average Pooling makes the models sensitive to global disease patterns as opposed to pixels. However, we also admit that extreme lighting and extreme occlusions are not completely covered by the existing dataset. Future research will thus involve the systematic assessment of the unedited field images and further data collection in more adverse conditions in order to further confirm the generalization in practice in the farm settings.
Data collection was conducted using a workstation equipped with an Nvidia Quadro P2000 GPU with 5GB, 32GB of RAM, and an Intel(R) Xeon(R) CPU E5-1650 running at 3.60 GHz. By employing the proposed separable-CNN model with Global Average Pooling, which has been trained on extensive datasets and incorporates learned weights and architecture, we can now integrate learning into our problem statement. A critical aspect of deploying deep learning models, particularly in real-world and resource-constrained environments, is the trade-off between model complexity and computational efficiency.
Table 4 compares various deep learning architectures, including both transfer learning models and proposed custom models, and reveals substantial differences in terms of the model size, number of parameters, and architectural depth. Overall, this comparative analysis underscores the advantages of the proposed lightweight models, particularly the Fast and Enhanced CNN variants, which significantly reduce the computational costs without compromising accuracy. These models are particularly well-suited for deployment in edge devices and field-based agricultural systems, where memory and processing resources are constrained.
Although the current research is based on the popular architectures, including VGG, ResNet, Xception, and Inception families as the baseline, recent lightweight-based models, like MobileNetV3 and EfficientNet, are also good candidates to use on-device. An empirical comparison of these architectures systematically with these architectures, with the same soybean data set and an identical evaluation protocol, would be an interesting future research direction, particularly when it comes to deployments on ultra-low power hardware platforms.

Table 4: Models Deep Learning Architectures

Model	Size	Parameters	Depth
Xception	88 MB	22,910,480	126
VGG16	528 MB	138,357,544	23
VGG19	549 MB	143,667,240	26
ResNet50	98 MB	25,636,712	-
InceptionV3	92 MB	23,851,784	159
InceptionResNetV2	215 MB	55,873,736	572
ResNet152V2	232 MB	60,380,648	-
Proposed CNN Models	1.24 GB	111,485,192	20
Proposed Fast CNN Models	9.04 MB	780,227	19
Proposed Enhanced CNN Models	23.4 MB	2,034,529	25

As shown in Figure 6(a), the Fast CNN model showed a steady increase in both training and validation accuracy, reaching over 96%, indicating effective learning and good generalization without overfitting. As shown in Figure 6(b), the training and validation losses decreased consistently over the epochs, with minimal fluctuation, confirming the Fast CNN's stability and strong convergence. As shown in Figure 6(c), the Enhanced CNN achieved high training and validation accuracy (above 97%), showing strong generalization and reliable model performance. As shown in Figure 6(d), both the training and validation losses steadily declined, with low final values, indicating efficient learning and minimal overfitting in the Enhanced CNN model.

The confusion matrix for the Fast CNN model in Figure 7(a) demonstrates a strong classification performance across all soybean disease classes. Most categories showed high true-positive rates, indicating highly accurate predictions. Misclassifications were minimal, with slightly more confusion observed between the visually similar classes. Overall, the matrix confirms the robustness and reliability of Fast CNN in distinguishing between multiple disease types and healthy samples, with only minor confusion in complex cases. The confusion matrix for the Enhanced CNN model, as depicted in Figure 7(b), demonstrates exceptional classification accuracy across all categories of soybean diseases. Most classes exhibited nearly perfect predictions. The overall diagonal dominance within the matrix underscores the model's excellent capability to distinguish between diseases, thereby confirming that the Enhanced CNN achieved highly reliable and precise disease detection.

An in-depth quantitative analysis was carried out on the Proposed Fast CNN illustrated in Figure 8(a) and the Enhanced CNN shown in Figure 8(b) to evaluate their classification performance using a 70-20-10 data split. The analysis focused on critical metrics: precision, recall, and F1-score for each disease class, along with macro and weighted averages and overall accuracy. The metrics offer valuable insights into the strength, equilibrium, and dependability of the models in differentiating among various soybean disease categories. The Fast CNN model attained an overall accuracy rate of 96%, demonstrating consistent and balanced performance across all disease classes. The Enhanced CNN model further enhanced classification performance, reaching a higher overall accuracy of 97%. This model also exhibited superior precision and recall in nearly all categories.

These results confirm that both proposed models are highly reliable for soybean disease classification. However, the Enhanced CNN exhibits a slight advantage in terms of precision and recall, making it the preferred model for applications demanding high diagnostic accuracy. On the other hand, Fast CNN remains a strong candidate for deployment scenarios requiring faster inference with minimal trade-off in accuracy.

As shown in Table 5, the performance of the proposed CNN architectures was benchmarked against several state-of-the-art deep learning models, including VGG16, VGG19, ResNet152V2, Xception, InceptionV3, and InceptionResNetV2, using different training-validation-testing data splits 70-20-10, 65-25-10, and 60-30-10. Across all configurations, the Proposed Enhanced CNN consistently outperformed the baseline models in terms of accuracy, achieving up to 98.14% test accuracy with the 70-20-10 split.

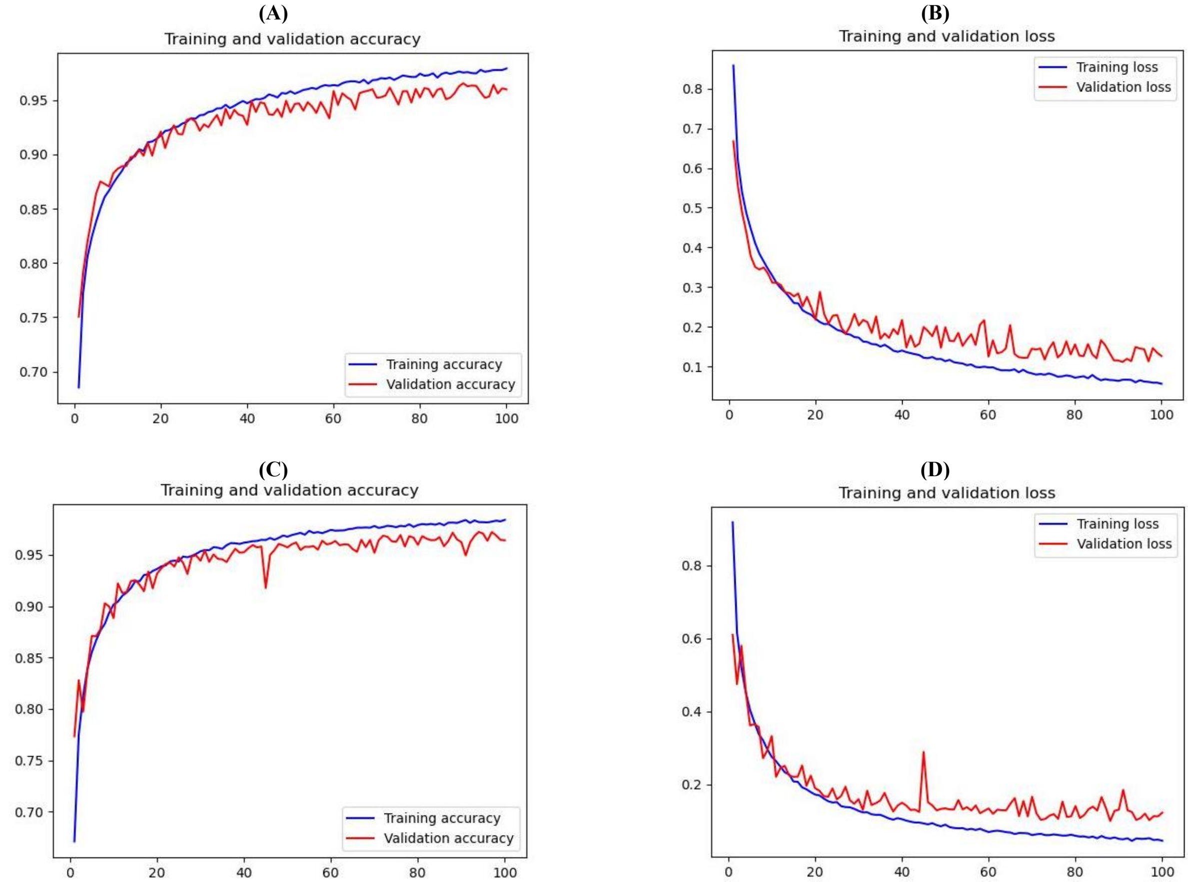

Fig. 6: Model Training (A) Fast CNN Training Accuracy Plot, (B) Fast CNN Training Loss Plot, (C) Enhanced CNN Training Accuracy Plot, (D) Enhanced CNN Training Loss Plot.

(A) (B)

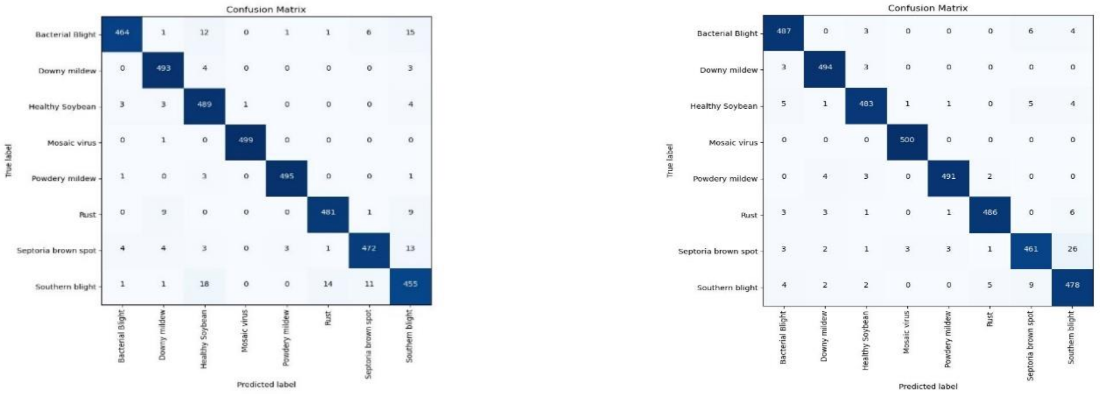

Fig. 7: Confusion Matrix (A) Fast CNN (B) Enhanced CNN.

(A)

```
Classification Report
                     precision    recall  f1-score   support

    Bacterial_Blight       0.98      0.93      0.95       500
        Downy_mildew       0.96      0.99      0.97       500
     Healthy_Soybean       0.92      0.98      0.95       500
        Mosaic_virus       1.00      1.00      1.00       500
      Powdery_mildew       0.99      0.99      0.99       500
                Rust       0.97      0.96      0.96       500
 Septoria_brown_spot       0.96      0.94      0.95       500
     Southern_blight       0.91      0.91      0.91       500

            accuracy                           0.96      4000
           macro avg       0.96      0.96      0.96      4000
        weighted avg       0.96      0.96      0.96      4000
```

(B)

```
Classification Report
                     precision    recall  f1-score   support

    Bacterial_Blight       0.96      0.97      0.97       500
        Downy_mildew       0.98      0.99      0.98       500
     Healthy_Soybean       0.97      0.97      0.97       500
        Mosaic_virus       0.99      1.00      1.00       500
      Powdery_mildew       0.99      0.98      0.99       500
                Rust       0.98      0.97      0.98       500
 Septoria_brown_spot       0.96      0.92      0.94       500
     Southern_blight       0.92      0.96      0.94       500

            accuracy                           0.97      4000
           macro avg       0.97      0.97      0.97      4000
        weighted avg       0.97      0.97      0.97      4000
```

Fig. 8: Evaluation (A) Fast CNN (B) Enhanced CNN.

Table 5: Models Comparative Analysis

For Validation Data						
	70-20-10		65-25-10		60-30-10	
Model	loss	accuracy	loss	accuracy	loss	accuracy
VGG16	0.80435	0.73242	0.69931	0.76074	0.79009	0.73242
VGG19	0.78891	0.71484	0.89637	0.70117	0.67836	0.75684
ResNet152V2	0.95172	0.82031	1.37763	0.74023	1.44067	0.74707
Xception	0.89843	0.72461	0.80814	0.74805	1.09496	0.68359
InceptionV3	0.74500	0.76660	0.84093	0.74414	1.07984	0.71094
InceptionResNetV2	0.74384	0.78711	0.58134	0.82129	0.66821	0.79688
Proposed CNN	0.09773	0.96582	0.09273	0.96972	0.16907	0.95214
Proposed Fast CNN	0.11767	0.95605	0.10096	0.96582	0.12199	0.96386
Proposed Enhanced CNN	0.11547	0.96093	0.11029	0.96777	0.12348	0.96093
For Test Data						
	70-20-10		65-25-10		60-30-10	
Model	loss	accuracy	loss	accuracy	loss	accuracy
VGG16	0.50096	0.82813	0.42003	0.85742	0.47544	0.83496
VGG19	0.51112	0.82422	0.55682	0.79883	0.48989	0.82129
ResNet152V2	0.62186	0.87598	0.89074	0.82617	0.85621	0.83008
xception	0.55212	0.83496	0.48634	0.85645	0.71710	0.79590
InceptionV3	0.53964	0.83789	0.65395	0.81348	0.60704	0.80469
InceptionResNetV2	0.57007	0.84375	0.40065	0.87207	0.52754	0.83887
Proposed CNN	0.20552	0.94824	0.19249	0.95019	0.23364	0.93164
Proposed Fast CNN	0.14723	0.95703	0.15041	0.95605	0.17079	0.95312
Proposed Enhanced CNN	0.07590	0.98144	0.14053	0.96875	0.12608	0.97558

Table 6: Time Complexity Analysis

	70-20-10 Data Ratio	65-25-10 Data Ratio	60-30-10 Data Ratio
Models	(Hours)	(Hours)	(Hours)
Proposed CNN 100 E	08:20:30	07:55:01	08:44:57
Proposed Fast CNN 100 E	09:12:56	09:01:42	08:12:56
Proposed Enhanced CNN 100 E	18:43:32	17:21:02	17:04:02

Notably, the Proposed Fast CNN also delivered impressive results, balancing a high accuracy (95.70%) with a lower training time compared to the enhanced model.

In comparison, traditional models, such as VGG16 and VGG19, yielded lower accuracies (ranging from 73% to 86%), and even more advanced models, such as ResNet152V2 and InceptionV3, did not surpass 88%, indicating that the proposed architectures provide significant improvements. Moreover, the proposed models demonstrated substantially lower loss values, particularly in the validation and test phases, highlighting their superior generalization capabilities.

In terms of computational efficiency, the Proposed CNN required the least training time (~8 h), whereas the Enhanced CNN, owing to its more complex architecture, required almost double the time (~17–18 h). The Fast CNN offered a middle ground, maintaining competitive accuracy with reasonable training time, making it ideal for real-time or resource-constrained applications.

Overall, the comparative analysis clearly demonstrates that the proposed models, especially the Enhanced and Fast CNNs, significantly surpass existing state-of-the-art methods in terms of both accuracy and robustness, while offering flexibility in training time depending on the application needs.

In addition to the current experimental assessment, there are other prospects of future research. To begin with, the suggested Fast and Enhanced CNN models can be readily incorporated into edge-computing pipelines, like smartphone applications to aid farmers, cheap embedded systems deployed in the field, and IoT nodes with a connection to the farm management system. On-device inference would, in such cases, allow disease alerts in near real-time without the need to be connected to cloud servers all the time. Second, integrating the models with unmanned aerial vehicles (UAVs) or drones with RGB or multispectral cameras may enable large-scale monitoring of the canopy and early diversity outbreaks in large soybean fields. Third, the framework can be expanded to multi-crop and multi-modal environments (including other sensor data such as weather, soil moisture, or hyperspectral images), and other domain adaptation methods can be considered to become more robust in a variety of different agroclimatic conditions. Lastly, systematic comparisons to recent lightweight families like MobileNetV3 and EfficientNet, along with model compression and quantization techniques, will be explored in future research so that they can be deployed to ultra-low power IoT platforms.

5. Conclusion

This study introduced a fast and enhanced Convolutional Neural Network (CNN) approach for detecting soybean leaf diseases in Indian agriculture. The proposed model combines a Separable CNN with a Global Average Pooling (GAP) layer to reduce the computational complexity while maintaining high accuracy. The model was trained on a comprehensive dataset of soybean leaf disease categories common in India. The experimental results demonstrated the superiority of the Separable-CNN with the GAP model over traditional CNN methodologies, achieving up to 98.14% test accuracy. The Enhanced CNN model outperformed several state-of-the-art architectures such as VGG16, ResNet152V2, and InceptionV3. The Fast CNN variant provides a balance between high accuracy and reduced training time, making it suitable for real-time applications. The confusion matrix analysis confirmed that the proposed models accurately classified most disease categories with minimal misclassification. These results validate the effectiveness of the proposed architectures in delivering high-precision plant disease detection. The proposed approach has significant implications for improving disease management in agriculture through early and accurate detection. This paper is part of the emerging area of deep learning applications in agriculture and evidence of how advanced but simplified CNN models can be used to support scalable real-time monitoring of diseases by integrating with mobile devices, IoT systems, and drone-based imaging systems.

References

[1] E. D. K. Ruby, S. Gopalakrishnan, T. Chitra, A. Choubey, G. Amirthayogam, and G. Sasi, “Advanced Image Processing Techniques for Automated Detection of Healthy and Infected Leaves in Agricultural Systems,” *Mesopotamian Journal of Computer Science*, vol. 2024, pp. 62–70, Jun. 2024, https://doi.org/10.58496/MJCSC/2024/006.

[2] A. Haridasan, J. Thomas, and E. D. Raj, “Deep learning system for paddy plant disease detection and classification.” *Environ Monit Assess*, vol. 195, no. 1, Nov. 2022, https://doi.org/10.1007/s10661-022-10656-x.

[3] A. B. Duque, D. Macêdo, C. Zanchettin, and L. L. J. Santos, “Squeezed Very Deep Convolutional Neural Networks for Text Classification,” Springer, 2019, pp. 193–207. https://doi.org/10.1007/978-3-030-30487-4_16.

[4] B. Liu, J. Li, D. Zou, L. Feng, P. Fu, and S. Feng, “An FPGA-Based CNN Accelerator Integrating Depthwise Separable Convolution,” *Electronics (Basel)*, vol. 8, no. 3, p. 281, Mar. 2019, https://doi.org/10.3390/electronics8030281.

[5] Q. Zhu, X. Jiang, and R. Ye, “Sentiment Analysis of Review Text Based on BiGRU-Attention and Hybrid CNN,” *IEEE Access*, vol. 9, pp. 149077–149088, Jan. 2021, https://doi.org/10.1109/ACCESS.2021.3118537.

[6] H. Sinha, V. Awasthi, and P. K. Ajmera, “Audio classification using braided convolutional neural networks,” *IET Signal Processing*, vol. 14, no. 7, pp. 448–454, Sep. 2020, https://doi.org/10.1049/iet-spr.2019.0381.

[7] V. Christlein, M. Seuret, A. Nicolaou, L. Spranger, A. Maier, and P. Kral, “Deep Generalized Max Pooling,” Institute of Electrical Electronics Engineers, Sep. 2019, pp. 1090–1096. https://doi.org/10.1109/ICDAR.2019.00177.

[8] T. Liu *et al.*, “Spatial Channel Attention for Deep Convolutional Neural Networks,” *Mathematics*, vol. 10, no. 10, p. 1750, May 2022, https://doi.org/10.3390/math10101750.

[9] P. Vasavi, A. Punitha, and T. V. Narayana Rao, “Crop leaf disease detection and classification using machine learning and deep learning algorithms by visual symptoms: a review,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, p. 2079, Apr. 2022, https://doi.org/10.11591/ijece.v12i2.pp2079-2086.

[10] L. W. Kuswidiyanto, X. Han, and H.-H. Noh, “Plant Disease Diagnosis Using Deep Learning Based on Aerial Hyperspectral Images: A Review,” *Remote Sens (Basel)*, vol. 14, no. 23, p. 6031, Nov. 2022, https://doi.org/10.3390/rs14236031.

[11] H. Paul, H. Udayangani, K. Umesha, N. Lankasena, C. Liyanage, and K. Thambugala, “Maize leaf disease detection using a convolutional neural network: a mobile application based on pre-trained VGG16 architecture,” *N Z J Crop Hortic Sci*, vol. 53, no. 2, pp. 367–383, Aug. 2024, https://doi.org/10.1080/01140671.2024.2385813.

[12] C. P. Vijay and K. Pushpalatha, “DV-PSO-Net: A novel deep mutual learning model with Heuristic search using Particle Swarm optimization for Mango leaf disease detection,” *Journal of Integrated Science and Technology*, May 2024, https://doi.org/10.62110/sciencein.jist.2024.v12.804.

[13] H. Hairani and T. Widiyaningtyas, “Augmented Rice Plant Disease Detection with Convolutional Neural Networks,” *INTENSIF: Jurnal Ilmiah Penelitian dan Penerapan Teknologi Sistem Informasi*, vol. 8, no. 1, pp. 27–39, Feb. 2024, https://doi.org/10.29407/intensif.v8i1.21168.

[14] S. B. Jadhav, V. R. Udupi, and S. B. Patil, “Identification of plant diseases using convolutional neural networks,” *International Journal of Information Technology*, vol. 13, no. 6, pp. 2461–2470, Feb. 2020, https://doi.org/10.1007/s41870-020-00437-5.

[15] L. M. Abouelmagd, M. Y. Shams, H. S. Marie, and A. E. Hassanien, “An optimized capsule neural network for tomato leaf disease classification,” *EURASIP J Image Video Process*, vol. 2024, no. 1, Jan. 2024, https://doi.org/10.1186/s13640-023-00618-9.

[16] E. Hirani, V. Magotra, P. Bide, and J. Jain, “Plant Disease Detection Using Deep Learning,” institute of electrical electronics engineers, Apr. 2021, pp. 1–4. https://doi.org/10.1109/I2CT51068.2021.9417910.

[17] A. Tuncer, “Cost-optimized hybrid convolutional neural networks for detection of plant leaf diseases,” *J Ambient Intell Humaniz Comput*, vol. 12, no. 8, pp. 8625–8636, May 2021, https://doi.org/10.1007/s12652-021-03289-4.

[18] M. S. Islam *et al.*, “Multimodal Hybrid Deep Learning Approach to Detect Tomato Leaf Disease Using Attention Based Dilated Convolution Feature Extractor with Logistic Regression Classification.,” *Sensors*, vol. 22, no. 16, p. 6079, Aug. 2022, https://doi.org/10.3390/s22166079.

[19] V. K. Vishnoi, B. Kumar, K. Kumar, A. A. Khan, and S. Mohan, “Detection of Apple Plant Diseases Using Leaf Images Through Convolutional Neural Network,” *IEEE Access*, vol. 11, pp. 6594–6609, Jan. 2023, https://doi.org/10.1109/ACCESS.2022.3232917.

[20] M. Z. Hasan, K. M. Z. Hasan, M. S. Ahamed, and A. Rakshit, “Recognition of Jute Diseases by Leaf Image Classification using Convolutional Neural Network,” institute of electrical electronics engineers, Jul. 2019, pp. 1–5. https://doi.org/10.1109/ICCCNT45670.2019.8944907.

[21] A. Khan, A. Sohail, A. S. Qureshi, and U. Zahoora, "A survey of the recent architectures of deep convolutional neural networks," *Artif Intell Rev*, vol. 53, no. 8, pp. 5455–5516, Apr. 2020, https://doi.org/10.1007/s10462-020-09825-6.
[22] E. Suryawati, R. Sustika, R. S. Yuwana, H. F. Pardede, and A. Subekti, "Deep Structured Convolutional Neural Network for Tomato Diseases Detection," institute of electrical electronics engineers, Oct. 2018, pp. 385–390. https://doi.org/10.1109/ICACSIS.2018.8618169.
[23] R. Tabares-Soto *et al.*, "Strategy to improve the accuracy of convolutional neural network architectures applied to digital image steganalysis in the spatial domain.," *PeerJ Comput Sci*, vol. 7, no. 5, p. e451, Apr. 2021, https://doi.org/10.7717/peerj-cs.451.
[24] M. Dai *et al.*, "Pepper leaf disease recognition based on enhanced lightweight convolutional neural networks.," *Front Plant Sci*, vol. 14, Aug. 2023, https://doi.org/10.3389/fpls.2023.1230886.
[25] A. Y. Ashurov *et al.*, "Enhancing plant disease detection through deep learning: a Depthwise CNN with squeeze and excitation integration and residual skip connections.," *Front Plant Sci*, vol. 15, Jan. 2025, https://doi.org/10.3389/fpls.2024.1505857.
[26] T. S. Alam, C. B. Jowthi, and A. Pathak, "Comparing pre-trained models for efficient leaf disease detection: a study on custom CNN," *Journal of Electrical Systems and Information Technology*, vol. 11, no. 1, Feb. 2024, https://doi.org/10.1186/s43067-024-00137-1..
[27] A. Zisserman and K. Simonyan, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014.
[28] K. He, S. Ren, X. Zhang, and J. Sun, "Identity Mappings in Deep Residual Networks," Mar. 2016, *cornell university*. https://doi.org/10.1007/978-3-319-46493-0_38.
[29] Y. Liu, S. Xu, D. Cheng, J. Han, and D. Zhang, "Capsule Networks with Residual Pose Routing.," *IEEE Trans Neural Netw Learn Syst*, vol. 36, no. 2, pp. 2648–2661, Feb. 2025, https://doi.org/10.1109/TNNLS.2023.3347722.
[30] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Oct. 2016, *cornell university*. https://doi.org/10.1109/CVPR.2017.195.
[31] R. Zhao, W. Luk, and X. Niu, "Automatic Optimising CNN with Depthwise Separable Convolution on FPGA," association for computing machinery, Feb. 2018, p. 285. https://doi.org/10.1145/3174243.3174959.
[32] E. Rahimian, S. F. Atashzar, S. Zabihi, A. Asif, and A. Mohammadi, "XceptionTime: Independent Time-Window Xceptiontime Architecture for Hand Gesture Classification," institute of electrical electronics engineers, May 2020, pp. 1304–1308. https://doi.org/10.1109/ICASSP40776.2020.9054586.
[33] C. Szegedy, Z. Wojna, J. Shlens, S. Ioffe, and V. Vanhoucke, "Rethinking the Inception Architecture for Computer Vision," institute of electrical electronics engineers, Jun. 2016, pp. 2818–2826. https://doi.org/10.1109/CVPR.2016.308.
[34] M. Z. Alom, C. Yakopcic, V. K. Asari, M. Hasan, and T. M. Taha, "Inception recurrent convolutional neural network for object recognition," *Mach Vis Appl*, vol. 32, no. 1, Jan. 2021, https://doi.org/10.1007/s00138-020-01157-3.
[35] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, Feb. 2017, https://doi.org/10.1609/aaai.v31i1.11231.
[36] M. G. Ragab, S. J. Abdulkadir, and N. Aziz, "Random Search One Dimensional CNN for Human Activity Recognition," institute of electrical electronics engineers, Oct. 2020, pp. 86–91. https://doi.org/10.1109/ICCI51257.2020.9247810.
[37] G. Lu, Z. Wang, and W. Zhang, "Optimizing Depthwise Separable Convolution Operations on GPUs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 1, pp. 70–87, Jan. 2022, https://doi.org/10.1109/TPDS.2021.3084813.
[38] R. Zhao, W. Luk, and X. Niu, "Automatic Optimising CNN with Depthwise Separable Convolution on FPGA," association for computing machinery, Feb. 2018, p. 285. https://doi.org/10.1145/3174243.3174959.
[39] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," Oct. 2016, *cornell university*. https://doi.org/10.1109/CVPR.2017.195.
[40] R. Ye, L. Zhang, and F. Liu, "3D Depthwise Convolution: Reducing Model Parameters in 3D Vision Tasks," springer, 2019, pp. 186–199. https://doi.org/10.1007/978-3-030-18305-9_15.
[41] M. Z. Asghar *et al.*, "Facial Mask Detection Using Depthwise Separable Convolutional Neural Network Model During COVID-19 Pandemic.," *Front Public Health*, vol. 10, Mar. 2022, https://doi.org/10.3389/fpubh.2022.855254.
[42] A. Bhujel, N.-E. Kim, E. Arulmozhi, J. K. Basak, and H.-T. Kim, "A Lightweight Attention-Based Convolutional Neural Networks for Tomato Leaf Disease Classification," *Agriculture*, vol. 12, no. 2, p. 228, Feb. 2022, https://doi.org/10.3390/agriculture12020228.
[43] Y. Gulzar, "Enhancing soybean classification with modified inception model: A transfer learning approach," *Emir J Food Agric*, vol. 36, pp. 1–9, Apr. 2024, https://doi.org/10.3897/ejfa.2024.122928.
[44] L. Meng, "The Convolutional Neural Network Text Classification Algorithm in the Information Management of Smart Tourism Based on Internet of Things," *IEEE Access*, vol. 12, pp. 3570–3580, Jan. 2024, https://doi.org/10.1109/ACCESS.2024.3349386.
[45] E. Westphal and H. Seitz, "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks," *Addit Manuf*, vol. 41, p. 101965, Mar. 2021, https://doi.org/10.1016/j.addma.2021.101965.