ORIGINAL RESEARCH **Open access**

Lite-MDC: A Lightweight Deep Learning Model for Real-Time Plant Disease Detection on Edge Devices

Daniel Fischer[1*], Laura Meier[1], Stefan Koch[2], Thomas Braun[3]

Abstract

Identifying plant diseases early and initiating timely treatment are crucial to achieving sustainable crop production. The field of computer vision applied to crop science is expanding quickly alongside major advances in deep learning. However, real-time plant disease detection remains difficult owing to the erratic progression of disease across plant parts, varying environmental conditions, and limited availability of authentic field-collected datasets. This study addresses these challenges through three essential elements: (a) jointly developing a new pigeon pea image dataset sourced directly from farm fields, in collaboration with 20 Agricultural Research Centers (ARS) and government bodies across 18 Indian states. (b) Creating compact yet powerful models capable of real-time plant disease detection on devices with limited resources. (c) Capturing multi-scale features of plant diseases through Multi-kernel Depthwise separable Convolutions. The suggested lightweight Lite-MDC architecture incorporates Multi-kernel Depthwise separable Convolutions (MDsConv). This MDsConv component gathers spatial information from multiple scales without increasing model size. It successfully extracts multi-scale details needed to represent plant diseases of different sizes. The suggested design greatly reduces computational demands, using only 2.2 million parameters — a 62% decrease relative to the standard VGG16 architecture. On the custom pigeon pea dataset, the suggested technique outperforms leading networks, including InceptionV3, VGG16, ResNet50, DenseNet, MobileNet, MobileNetV3, NASNet, and EfficientNetB0, achieving 94.14% accuracy. Remarkably, it delivers 34 Frames Per Second (FPS) during inference on an NVIDIA P100 GPU. In addition, its effectiveness was confirmed using open datasets, including the PlantVillage, Cassava, and apple leaf datasets, achieving 99.78%, 86.4%, and 97.2% accuracy, respectively. The Lite-MDC model shows strong promise for real-time plant disease detection on edge devices with limited resources, including agricultural robots and drones.

Keywords Computer vision, Disease detection, Depthwise separable convolution, Efficient CNN

*Correspondence:
Daniel Fischer
daniel.fischer@uni-freiburg.de
[1] Department of AI for Rapeseed and Mustard Green Manure Planning, Faculty of Agriculture, University of Freiburg, Freiburg, Germany
[2] Department of Smart Crop Rotation Scheduling Algorithms, Faculty of Agriculture, Technical University of Munich, Munich, Germany
[3] Department of Machine Learning for Biomass Estimation, Faculty of Agriculture, University of Kiel, Kiel, Germany

Introduction

Worldwide, pigeon pea ranks sixth among legume crops in Asian countries [1]. It is an excellent source of protein, high in fiber, and rich in essential amino acids. Nevertheless, frequent plant disease outbreaks can severely damage plant vitality and lower overall productivity. Each year, plant diseases result in economic losses ranging from 12 to 23 million dollars [2, 3]. At present, identifying various plant diseases largely depends on manual visual checks. Growers and specialists typically examine plants by sight to detect issues. This traditional practice suffers from notable drawbacks. Personal

This is an open access article distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (CC BY-NC-SA 4.0), which permits non-commercial use, distribution, reproduction, and adaptation in any medium or format, provided that appropriate credit is given to the original author(s) and source, a link to the license is provided, and any changes made are indicated. Adapted material must be distributed under the same license.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

biases and eye strain often cause unreliable judgments [4]. Incorrect disease diagnosis can interfere with proper monitoring and management efforts, leading to unsuitable pesticide application, poorer crop quality, and lower harvests [5]. Given the serious consequences of misdiagnosing plant diseases, accurate identification is essential for protecting plant health and achieving maximum yields.

Deep learning is a sophisticated technology for plant disease detection that automatically recognizes symptoms and patterns [6]. Yet it faces several difficulties, such as (a) the availability of real field datasets, (b) randomly distributed plant diseases on leaves, stems, and roots, and their diversity in texture, size, and shape, (c) environmental challenges include image blur due to wind, leaf overlapping, complex background, and viewpoints, as shown in **Figure 1**, and (d) the development of lightweight yet high-performance models tailored for real-time plant disease detection on resource-constrained embedded devices poses another significant challenge. Convolutional neural networks (CNNs) have emerged as a strong candidate for addressing challenges in disease recognition and categorization. These CNN designs are specifically designed for image processing and can overcome the shortcomings of conventional manual inspection. Even so, such models require significant computational resources. For instance, the standard VGG16 architecture consists of 92 layers and requires 140M parameters [7, 8]. To overcome this drawback, we introduced the Lite-MDC architecture based on multi-kernel depthwise convolutions. One major benefit of the proposed architecture is preserving top-level performance while reducing computational requirements. This strategy improves model efficiency without compromising precision. This paper also presents the first real-world dataset of pigeon pea images. The main contributions of this research are outlined below:

1. The proposed first novel pigeon-pea disease image dataset includes high-resolution images collected from 18 geographical locations across India.

2. The proposed MDsConv block with 3×3 and 5×5 kernels helps extract multi-scale features of plant diseases.

3. Proposed a lightweight architecture, Lite-MDC, for pigeon pea disease detection, which reduces computational complexity and shows state-of-the-art results with only 2.2 M parameters, with 34 FPS, suitable for real-time application.

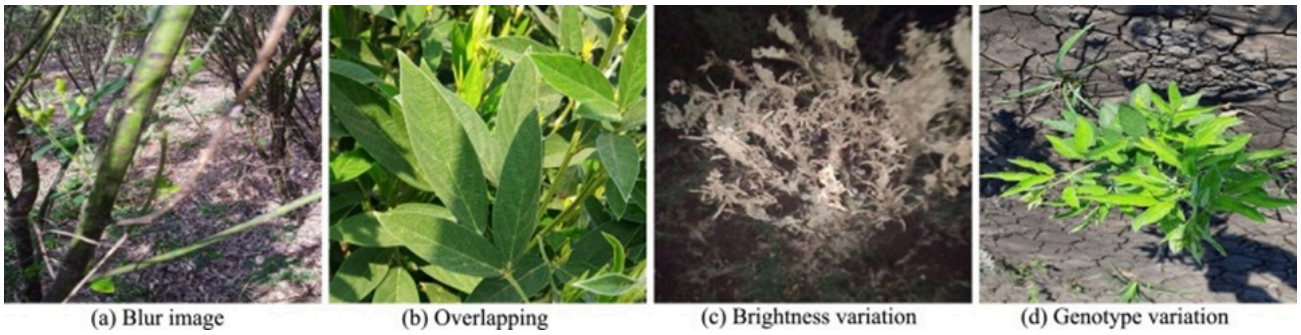


Figure 1. Challenges for pigeon pea disease detection in the pigeon pea dataset.

To assess how well the suggested method works, comprehensive tests were performed comparing it with public datasets and prominent existing techniques such as VGG-16 [9], Xceptions [10], NasNet mobile [11], MobileNet [12], MobileNetV2 [13], ResNet50 [14], InceptionV3 [15], DenseNet121 [16], and EfficientNetB0 [17]. This thorough assessment provides a solid measure of the model's capabilities and demonstrates its clear benefits over alternative solutions.

The remainder of the paper is organized as follows. Section 2 reviews the literature. Section 3 presents the details of constructing the pigeonpea dataset; Section 4 introduces the proposed method for plant disease detection. Section 5 enumerates the experimental details, and Section 6 illustrates the results and analysis with discussion. Finally, Section 7 concludes the work by outlining some future directions.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Related Work

In recent years, the agricultural sector has seen growing interest in detecting and managing plant diseases, driven by advances in image processing and machine learning.

Image processing and machine learning for plant disease detection

Image processing combined with machine learning methods has been widely adopted to identify diseases in multiple crops, such as tomatoes, potatoes, sugarcane, and maize [18-20]. These approaches involve several image processing strategies, including hand-crafted features, scale-invariant feature transform, histogram-based techniques, thresholding, and segmentation algorithms [21-24]. Hlaing and Zaw [25] applied statistical texture and color features to build an improved version of the Scale Invariant Feature Transform (SIFT) for tomato plant disease classification. They enhanced the richness of the SIFT descriptor by incorporating additional statistical elements while simultaneously reducing data dimensionality. Chaudhary *et al.* [21] introduced a color transformation method for detecting disease areas that relies on the CIELAB color space and binary thresholding to separate affected regions. This technique struggles to detect diseased areas that share leaf-vein-like coloration. Patil *et al.* [22] presented an approach for estimating the severity of sugarcane diseases by using the triangle thresholding technique to accurately segment infected lesions. However, this method proves inadequate for precise lesion segmentation because of the diverse traits exhibited by different diseases. With ongoing advancements in machine learning across various fields, researchers have leveraged these developments to apply machine learning algorithms for plant disease identification [18, 26-28]. Al-Hiary *et al.* [26] employed Artificial Neural Networks (ANN) and K-means clustering for plant disease classification. The proposed system does not perform well on actual field-captured images. Sankaran and Ehsani [18] utilized visible-near-infrared and mid-infrared spectroscopy to classify leaves affected by huanglongbing and citrus canker. The classification was performed using quadratic discriminant analysis (QDA) and k-nearest neighbors (kNN). The KNN classifier is highly resource-intensive and requires keeping the complete training data in memory, particularly with large-scale datasets.

Additionally, Revathi *et al.* [19] extracted features such as skew divergence, color variance, edge information, and texture from infected leaves, then applied a deep forward neural network to classify cotton leaf spot diseases. This solution works effectively only under controlled laboratory conditions. Karmokar *et al.* [20] proposed a Neural Network Ensemble (NNE) technique that combines multiple features to detect tea leaf diseases. The proposed method is highly susceptible to changes in the appearance of disease on tea leaves. Waghmare *et al.* [27] developed a technique for detecting grape diseases using leaf texture analysis and pattern recognition. The texture features obtained were then classified using a multiclass Support Vector Machine (SVM).

Nevertheless, SVM performance tends to decline on noisy datasets. Machine learning models typically rely on manual feature engineering and are strongly influenced by the quality and quantity of the training data. Due to deep learning's advantages over conventional techniques, many researchers have shifted toward deep learning models for plant disease detection.

Deep learning for plant disease detection

Significant advances in deep learning for tasks such as image classification, segmentation, and object detection are revolutionizing this domain. In addition, deep learning finds extensive application in agriculture for monitoring crops, identifying diseases, estimating disease severity, predicting yields, and similar purposes [5, 6]. Mohanty *et al.* [5] employed the Plant Village dataset along with transfer learning using deep CNN models AlexNet and GoogLeNet to identify both plant species and associated diseases. However, AlexNet and GoogLeNet are resource-heavy models and unsuitable for deployment on edge devices. Kawasaki *et al.* [29] implemented a deep CNN using filter sizes of 12×12 and 5×5 to differentiate healthy cucumber leaves from infected ones. The research is restricted to a single dataset and overlooks factors such as sunlight exposure, leaf drooping, and other realistic field scenarios. Sladojevic *et al.* [30] used CaffeNet, a deep CNN variant of AlexNet, to classify plant diseases using web-sourced images as training data. This method achieved strong accuracy of 96.3% across 13 plant disease classes. Zhang *et al.* [31] applied transfer learning

© 2024 The Author(s). | Published by Agro Intelligence Research Press

with AlexNet, GoogleNet, and ResNet deep CNNs, reaching a peak accuracy of 97.28% in detecting tomato leaf diseases. Moreover, studies have expanded to include both disease classification and localization. Selvaraj *et al.* [32] adopted the single-shot detector (SSD) framework for banana disease identification, using MobileNetV1 as the backbone for feature extraction. Liu *et al.* [33] selected YOLOV3 as the base model for identifying tomato diseases and pests within the tomato dataset. The YOLOV3 model has 64M parameters, limiting its use for real-time processing on devices with limited resources. Wang *et al.* [34] trained and tested several CNN models on the PlantVillage dataset to assess the severity of apple black rot. In their work, the fine-tuned VGG16 model achieved 90.4% accuracy with 140M parameters. Biswas *et al.* [35] applied the VGG19 architecture to classify potato leaf conditions, including early blight, late blight, and healthy leaves, achieving 97.8% accuracy. In the study by Lu *et al.* [36], a Deep CNN model was introduced to categorize ten distinct rice leaf diseases using datasets of both healthy and diseased rice leaves. Although deep learning models demonstrate strong capabilities for plant disease detection, their high computational and memory requirements make them impractical for many real-world scenarios. This limitation can be addressed by developing efficient CNN architectures.

Efficient deep neural networks

Applications that require real-time performance on devices with limited resources require compact, lightweight CNN models [12, 13, 16, 37]. Standard CNN architectures like AlexNet [38], VGG [9], ResNet [39], GoogleNet [40], Inception ResNetV2 [41], among others, primarily focus on maximizing accuracy. Deploying such networks on edge devices remains difficult due to their substantial memory requirements, processing demands, and energy consumption. Refining these aspects enables the models to support real-time operations. Numerous studies have focused on developing efficient CNNs for image classification. Iandola *et al.* [37] introduced SqueezeNet, a compact CNN for image classification with just 1.25 million parameters. It delivered accuracy on par with AlexNet using 50 times fewer parameters [37]. In the study by Howard *et al.* [12], MobileNet was introduced as a lightweight CNN suitable for edge-device image classification. In related research, Sandler *et al.* [13] presented MobileNetV2, which outperforms MobileNet while using 4 M parameters. Zoph *et al.* [11] created NASNetMobile, a reduced version of NASNet intended for mobile and embedded vision systems. The current study introduces Lite-MDC, a lightweight multi-kernel depth-wise convolutional network inspired by VGG16 and multi-kernel depth-wise separable convolutions, making it well-suited for practical real-world use.

Datasets Used

This section introduces a proposed novel dataset construction for pigeonpea and various publicly available datasets used for comparison.

Pigeon-pea dataset construction

Insufficient data availability hinders the effective implementation of machine learning and deep learning strategies within agriculture. Challenges arising from environmental conditions and plant characteristics make it difficult to develop reliable, high-quality image collections. In response to these obstacles, the current study presents the inaugural custom pigeon-pea dataset. The complete workflow for building this dataset is shown in **Figure 2**. Creating the pigeon-pea dataset follows a structured three-step procedure: (1) system establishment, (2) dataset collection and annotation, and (3) dataset processing and verification.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

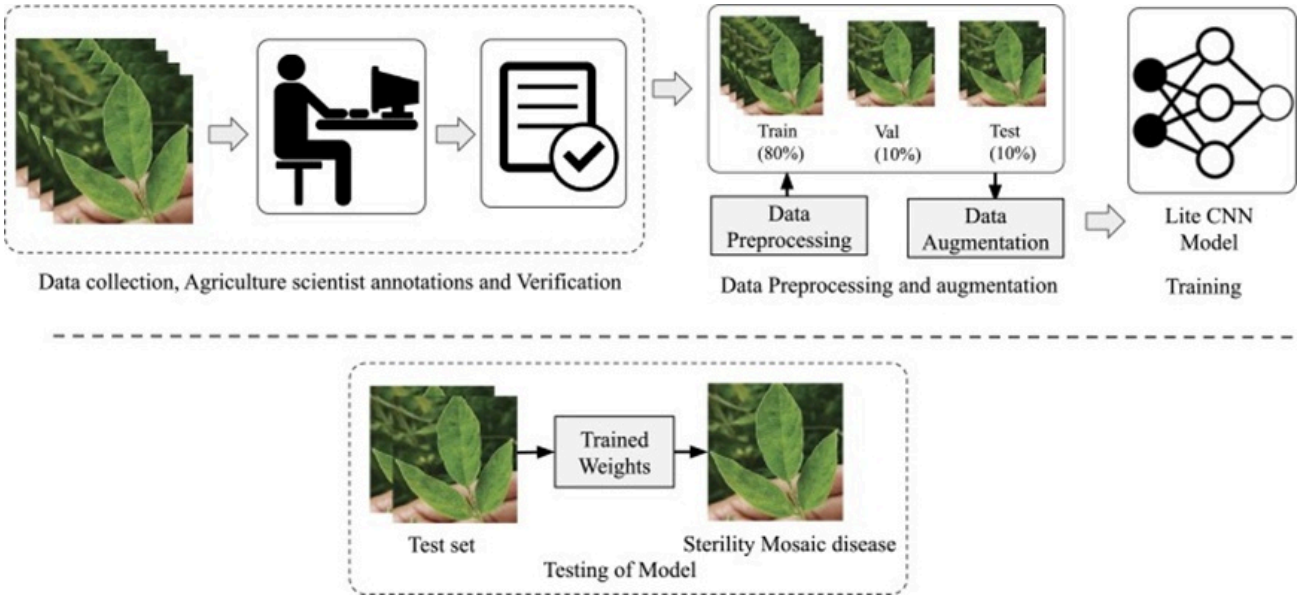


Figure 2. Data collection, annotation by experts, and data cleaning for novel pigeon-pea dataset.

System establishment

This effort commenced through the formal signing of a research partnership Memorandum of Understanding (MOU) linking our technical institute with an agricultural university. Supported by the Indian Council of Agricultural Research (ICAR) and the Indian Institute of Pulses Research (IIPR), 20 Agriculture Research Stations (ARSs) across 18 Indian states were identified. To ensure consistent image-acquisition standards across all ARSs, dedicated training was provided to agricultural scientists focused on photographing pigeon-pea disease.

Data collection and annotation

For generating a comprehensive pigeon-pea dataset, every participating station acquired photographs of four major pigeon-pea diseases: Wilt, Sterility mosaic, Dry root rot, Phytophthora blight, plus healthy specimens. Captures were performed with a range of smartphones and digital cameras, including the Samsung M21 and a Sony DSLR. Photography involved devices with varying resolutions, thereby incorporating a range of spatial resolutions throughout the collection. Agricultural scientists at each ARS manually labeled every image. Details on image counts per ARS and their distribution by category appear in **Tables 1 and 2**. The collection draws from 18 separate geographic sites illustrated in **Figure 3**, each presenting unique backgrounds and surrounding conditions. Photographs occurred across different daylight hours and at multiple disease manifestation scales, thereby strengthening dataset resilience and imparting originality. Examples highlighting variations in scale and background appear in **Figures 4 and 5**.

Table 1. Data collection from agriculture research stations (ARS) and pulse research stations across India.

Sr. No.	Crop	Images collection location	No. of images
1.	Pigeon-pea	Agriculture Research Station, Badnapur, Maharashtra	656
2.	Pigeon-pea	Agriculture Research Station, Ludhiyana, Punjab	600
3.	Pigeon-pea	Institute of Agricultural Science, Banaras Hindu University, Varanasi, Uttar Pradesh	820

© 2024 The Author(s). | Published by Agro Intelligence Research Press

4.	Pigeon-pea	Agriculture Research Station, Bengaluru, Karnataka	745
5.	Pigeon-pea	Agriculture Research Station, Coimbatore, Tamil Nadu	786
6.	Pigeon-pea	Agriculture Research Station, Warangal, Telangana	623
7.	Pigeon-pea	Agriculture Research Station, Tirupati, Andhra Pradesh	513
8.	Pigeon-pea	Agriculture Research Station, Guntur, Andhra Pradesh	704
9.	Pigeon-pea	Agriculture Research Station, Dharwad, Karnataka	565
10.	Pigeon-pea	Agriculture Research Station, Kanpur, Uttar Pradesh	790
11.	Pigeon-pea	Agriculture Research Station, Khargone, Madhya Pradesh	889
12.	Pigeon-pea	Agriculture Research Station, Vijayapura, Karnataka	827
13.	Pigeon-pea	Agriculture Research Station, Junagadh, Gujarat	640
14.	Pigeon-pea	Pulses Research Station, Sardar Krushinagar, India	700
15.	Pigeon-pea	International Crops Research Institute for the Semi-arid Tropics, Patancheru, India	875
16.	Pigeon-pea	Agriculture Research Station Rahuri, India	587
17.	Pigeon-pea	Indian Institute of Pulses Research, Kanpur, India	348
18.	Pigeon-pea	Pulses Research Station, Sardar Krushinagar, India	255
19.	Pigeon-pea	Pulses Research Station, Junagadh, Gujarat, India	336
20.	Pigeon-pea	Birsa Agricultural University, Ranchi	369
		Total	12,232

Table 2. Class-wise statistics for the proposed Pigeon-pea Dataset.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Classes	Collection condition	Number of images	Resolution	Annotations type
Dry root rot	Real field images	2103	1024 x 1024	Image level
Healthy	Real field images	2560	1024 x 1024	Image level
Phytophthora blight	Real field images	1691	1024 x 1024	Image level
Sterility Mosaic	Real field images	3652	1024 x 1024	Image level
Wilt	Real field images	2226	1024 x 1024	Image level
	Total	12,232		

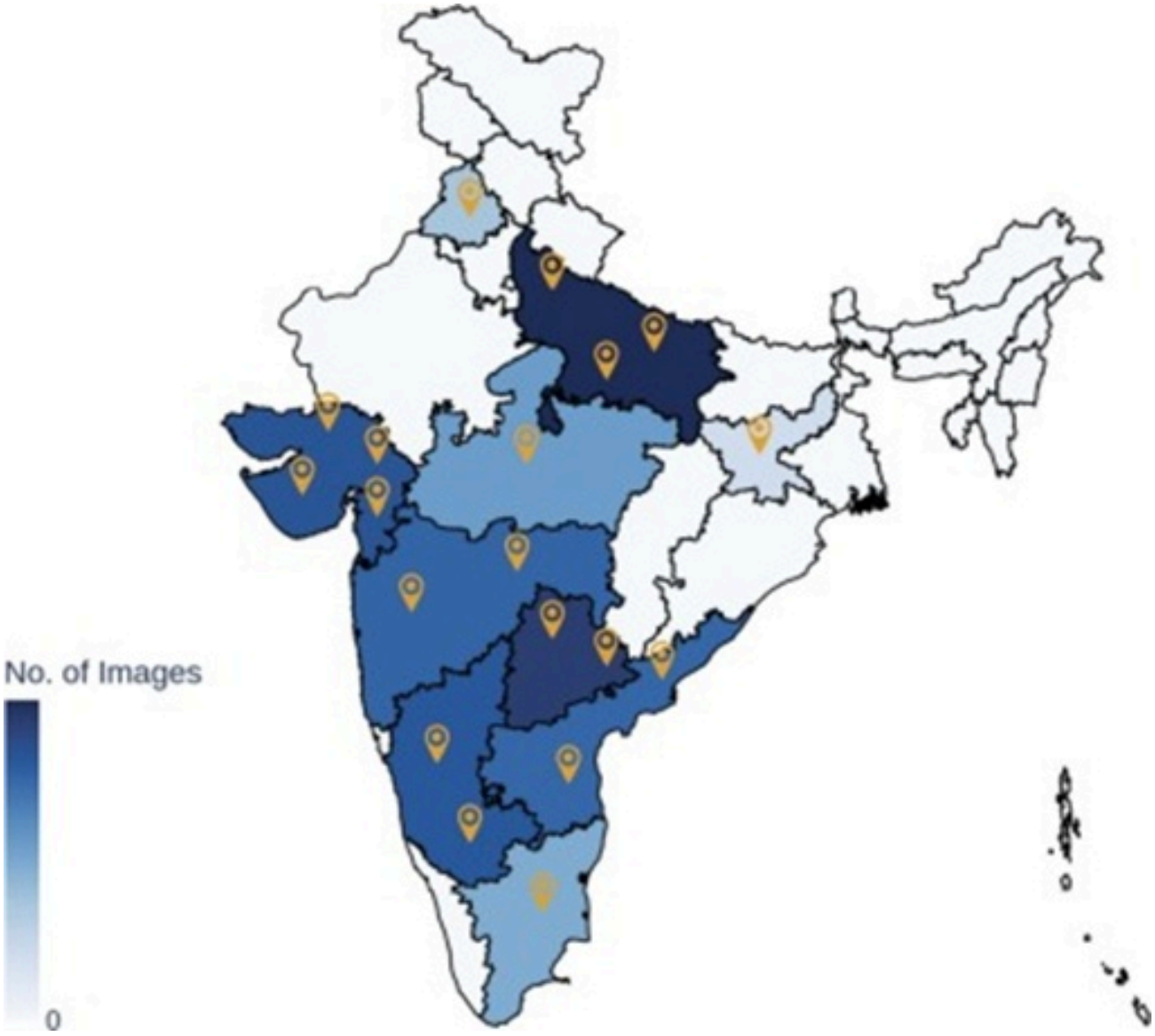


Figure 3. Data collection from twenty geographic locations, Agriculture Research Stations (ARS) and Pulse Research Stations across India.

© 2024 The Author(s). | Published by Agro Intelligence Research Press



Figure 4. Class-wise sample images from the proposed Pigeon-pea dataset.

© 2024 The Author(s). | Published by Agro Intelligence Research Press



Figure 5. Sample images from the Pigeon-pea dataset: (a) Sterility mosaic, (b) Wilt, (c) Phytophthora blight, (d) Dry root rot, (e) Healthy, and Multi-scale variation as shown in column (I) Closed, (II) Medium, and (III) Far.

Dataset processing and verification

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Once images were gathered, the collection entered a preparation stage that removed unclear or blurry entries. All remaining images were uniformly cropped to a spatial resolution of 1024×1024 to maintain consistent quality standards. Each image was then reviewed independently by computer vision specialists to correct potential inaccuracies. Before feeding into the CNN model, all images were resized to 224×224×3 via bilinear interpolation. Normalization centered the data by deducting each RGB channel's mean value from its corresponding pixels. The novel pigeon-pea dataset comprises 12,232 images across five classes representing typical pigeon-pea diseases. This Pigeon-pea dataset provides a comprehensive representation and variety of pigeon-pea diseases and remains openly accessible for research via GitHub link 1.

Publicly available datasets

Earlier investigations into plant disease recognition have drawn on multiple datasets, whose comprehensive statistics are presented in **Table 3**. Out of these collections, three stand as publicly obtainable: (1) Cassava Dataset [42], (2) the Plant Village dataset [6], and (3) the Apple dataset [43]. Evaluation of the proposed architecture is conducted across four datasets, including the fresh pigeon-pea dataset and three open-access datasets. Summaries of the datasets applied in this research are featured in **Table 4**. **Figure 6** displays examples of diseased and healthy specimens from each dataset.

Table 3. Statistics of the existing plant diseases dataset.

Dataset	Total images	Number of classes	Image collection environment	Disease location	Availability status
Leaflet Cassava Dataset [44]	1,896	6	Images captured in field conditions	Diseases present only on leaves	Not publicly available
Kaggle Cassava Dataset [42]	9,436	5	Field-acquired images	Diseases limited to leaves	Publicly accessible
Leaf Diseases Dataset [30]	4,483	15	Images collected from field environments	Only leaf diseases are included	Not available publicly
Plant Village Dataset [6]	54,309	38	Laboratory-generated images	Diseases affecting only leaves	Publicly available
Apple Foliar Disease Dataset [43]	3,651	4	Field-based image collection	Diseases confined to leaves	Publicly accessible
Apple Leaf Disease [45]	404	3	Images obtained under field conditions	Only leaf infections are considered	Not publicly available
Crop Pest Dataset [46]	4,483	40	Field images	Pest and disease symptoms on leaves only	Not available publicly
PDD271 [47]	220,592	271	Images gathered from field environments	Diseases restricted to leaves	Not publicly available
Pigeon-pea Dataset (Proposed)	12,232	5	Images captured in real field conditions	Diseases identified on the entire plant	Publicly available

Table 4. Details of datasets used for experimentation.

Dataset	Image acquisition condition	Number of classes	Total number of images	Image resolution

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Kaggle Cassava Dataset [42]	Images captured in real field environments	5	9,436	4000 × 2672
Apple Foliar Disease Dataset [43]	Real-world field images	4	3,651	1048 × 1365
PlantVillage Dataset [6]	Images collected under controlled indoor settings	38	54,309	256 × 256
Pigeon-pea Dataset (proposed)	Field-acquired real images	5	12,323	1024 × 1024
Total			**94,700**	



Figure 6. Sample images from (a) Pigeon-pea dataset, (b) Kaggle cassava dataset, (c) Plane Village dataset, and (d) Apple disease dataset.

Cassava dataset

Serving as the primary benchmark for cassava disease identification, the cassava dataset [42] draws on contributions from nearly 200 farmers working across varied Ugandan farmlands. Field images were obtained entirely through smartphone cameras. Composed of RGB images, this collection features five categories: healthy plant leaves together with four disease types — Cassava green mite (CGM), mosaic disease (CMD), bacterial blight (CBB), and brown streak disease (CBSD). Representative images from the set are shown in **Figure 6b**. Agriculture experts labeled 9,436 images possessing a resolution of 4000 × 2672. Relevant specifications are outlined in **Table 4**. Acquisition occurred through the "Cassava leaf disease competition" hosted on Kaggle. Given the absence of public ground truth for the test portion, the training data was split into training, validation, and test sets at 60%, 20%, and 20%, respectively.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

PlantVillage dataset

Widely recognized as a standard reference in plant disease detection, the PlantVillage dataset [6] contains 54,309 RGB images at 256 × 256 pixels each, with additional information in **Table 4**. Representing 38 unique plant disease types, all captures were conducted indoors under controlled conditions with uniform backgrounds. **Figure 6c** offers a visual summary. For experimental and assessment purposes, the full set was partitioned into training, validation, and test groups with allocations of 60%, 20%, and 20%, respectively.

Apple leaf dataset

The Apple leaf dataset [43] features diverse apple leaf disease images that vary in illumination, perspective, texture, and interference levels, originating from genuine outdoor field environments captured with a Canon Rebel T5i DSLR and smartphones. In total, it comprises 3,651 RGB images at a resolution of 1048 × 1365 pixels. Classification organizes these into four groups: apple scab, cedar apple rust, multiple diseases, and healthy leaves. Extended dataset information is presented in **Table 4**, with examples shown in **Figure 6d**. Sourced from the Plant Pathology Challenge on Kaggle, this collection lacks a public test set ground truth, prompting the division of the training data into training, validation, and test segments with 60%, 20%, and 20% proportions, respectively.

Proposed Methodology

This approach aims to build a streamlined network tailored to detecting plant diseases. Its central aim is to craft a resource-efficient CNN design that supports real-time operations and fits well on hardware with limited capabilities. Important attributes of effective CNN structures consist of (a) Lowered computational demands (overall count of multiplication and addition steps), (b) Reduced parameter quantity, (c) Decreased memory usage (model size), and (d) better results.

Convolutions in deep learning

Conventional convolutional layers form an essential foundation for deep learning systems, especially for analyzing images and videos. Still, its intensive processing needs can hamper suitability for live applications. Equations outlining the processing load and parameter totals for the conventional convolution feature in **Eq. 1** appear visually in **Figure 7a**. To address this restriction, depthwise convolution was introduced, applying unique kernels to each input channel, thereby reducing both parameter count and workload. Blending depthwise and pointwise (1x1) convolutions into depthwise separable convolutions (DWSC) is a common strategy for producing more optimized deep neural networks. Processing costs and parameter information for DWSC are supplied in **Eq. 2** and displayed in **Figure 7b**.

$$\begin{aligned} Cost_Std &= K \times K \times C \times N \times H \times W \\ Parm_Std &= K \times K \times C \times N + N \end{aligned} \quad (1)$$

Where, K×K = Kernel size, C = Number of input channels, N = Number of output channels, and H×W = Spatial Height and Width of input image or feature map (For simplicity, we consider padding = same, and stride = 0 and H=W).

$$\begin{aligned} Cost_DWsConv &= Cost_DW + Cost_PW \\ Cost_DWsConv &= (K \times K \times C_{in} \times H \times W) + \\ &(C_{in} \times C_{out} \times H \times W) \\ Parm_DWsConv &= (K \times K \times C_{in} + C_{in}) + (C_{in} \times N + N) \end{aligned} \quad (2)$$

© 2024 The Author(s). | Published by Agro Intelligence Research Press

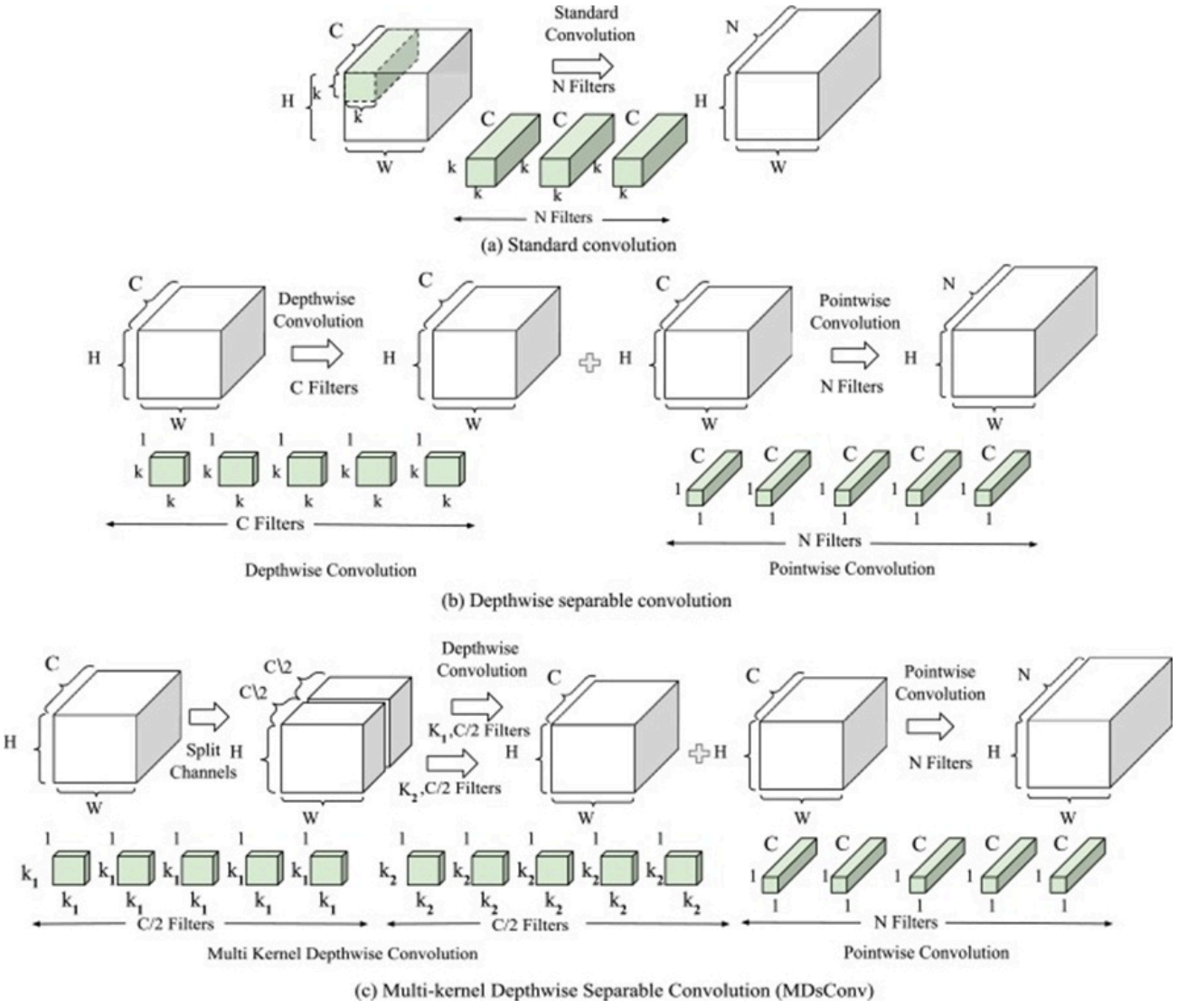


Figure 7. Convolution operations (a) Standard convolution, (b) Depthwise convolution, and (c) Multi-kernel Depthwise separable convolution (MDsConv).

Proposed multi-kernel depthwise separable convolution

While depthwise separable convolution delivers strong efficiency, it struggles when aggregating features at varying scales. To bridge this shortfall, we introduce Multi-kernel Depthwise separable Convolution (MDsConv). This method deploys a range of kernel sizes in the depthwise stage to capture multi-scale features from plant diseases. As a result, the multi-scale depthwise process facilitates understanding spatial links at multiple levels. Channels from the input are organized into groups, and the size of each group is determined by **Eq. 3**.

$$G_s = \frac{C_n}{K_i} \qquad (3)$$

Where Gs = Group size, Cn = Number of input channels, and Ki = Number of unique kernels. When Gs=1, MDConv is equivalent to standard depthwise convolution. Multiple kernels with depthwise convolution operations are applied to each channel group. The proposed MDsConv incorporates 3×3 and 5×5 kernels, resulting in a group size Gs=2, and splits the input channels into two groups. The number of 3×3 and 5×5 kernels equals half the number of channels in the input

© 2024 The Author(s). | Published by Agro Intelligence Research Press

feature map. The input feature map is divided into two groups as I = [$\frac{C}{2}$, $\frac{C}{2}$], where first $\frac{C}{2}$ feature map corresponding to 3×3 DwConv and the second $\frac{C}{2}$ feature map corresponding to 5×5 DWsConv is depicted in **Figure 7c**. Further, Pointwise convolution is used to fuse separable channels and to capture cross-channel dependencies. The complete MDsConv blocks are shown in **Figure 7c**. Complete pseudocode for MDsConv is given in **Algorithm 1**. The overall computational cost and parameters required for MDsConv are computed as follows:

$$
\begin{aligned}
&Cost_MDsConv = Cost_MDConv + Cost_PW \\
&Cost_MDsConv = \\
&\left\{(3 \times 3 \times \frac{C}{2} \times H \times W) + (5 \times 5 \times \frac{C}{2} \times H \times W)\right\} \\
&+ (C \times N \times H \times W) \\
&Parm_MDsConv = \left\{(3 \times 3 \times \frac{C}{2} + \frac{C}{2}) + (5 \times 5 \times \frac{C}{2} + \frac{C}{2})\right\} \\
&+ (C \times N + N)
\end{aligned}
\tag{4}
$$

Input: Input tensor (X_{in}), Kernel sizes [K_1, K_2]
Output: Output tensor X_{out}

```
G = len(X_in)
Kernels = [K_1, K_2]
Split = split(X_in, G, axis = −1)
Out = [ ]
# Multiscale depthwise convolution
for X_i, K_i in zip(Split, Kernels) do
    MDConv = DepthwiseConv2D(kernel_size = K_i)(X_i)
    Out.append(MDConv)
Out = Concatenate(Out)
Out = BatchNorm2D(Out)
# Pointwise convolution on concatenated output
Out = Conv2D(Out, kernel_size = 1)
X_out = BatchNorm2D(Out)
```

Algorithm 1. Multi-kernel depthwise separable Convolution (MDsConv).

Proposed lite-MDC architecture

Lite-MDC, the newly introduced framework, builds on a classic VGG16 architecture as its base for classifying plant diseases. To address shortcomings of the typical VGG16 setup, standard convolutional layers throughout were replaced with Multi-kernel Depthwise separable Convolution (MDsConv). An overview of the Lite-MDC framework is provided in

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Figure 8. Input images sized 224×224×3 enter the system and proceed through later stages dedicated to feature extraction. Batch normalization occurs after each layer in the Lite-MDC framework, allowing activations to adjust their scale to the incoming data. Such steps contribute toward higher precision levels and steadier behavior. Following batch normalization (except at the network's concluding layer), LeakyReLU functions as the activation. LeakyReLU addresses the problem of inactive neurons typical in ReLU by introducing a small slope for negative values, enabling faster training and improving overall performance. Feature map spatial reduction is achieved via a stride of 2 rather than max pooling. Dense layers are replaced by 1×1 pointwise convolutions with 512 filters, thereby reducing parameter counts while maintaining performance. Lastly, global average pooling is followed by a softmax stage that delivers final class predictions. The full layer breakdown is in **Table 5**. During training, dropout counters overfitting risks through sporadic neuron deactivation [48]. The dropout likelihood increases progressively from 0.05 to 0.4 near the final layers, as shown in **Figure 8**.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

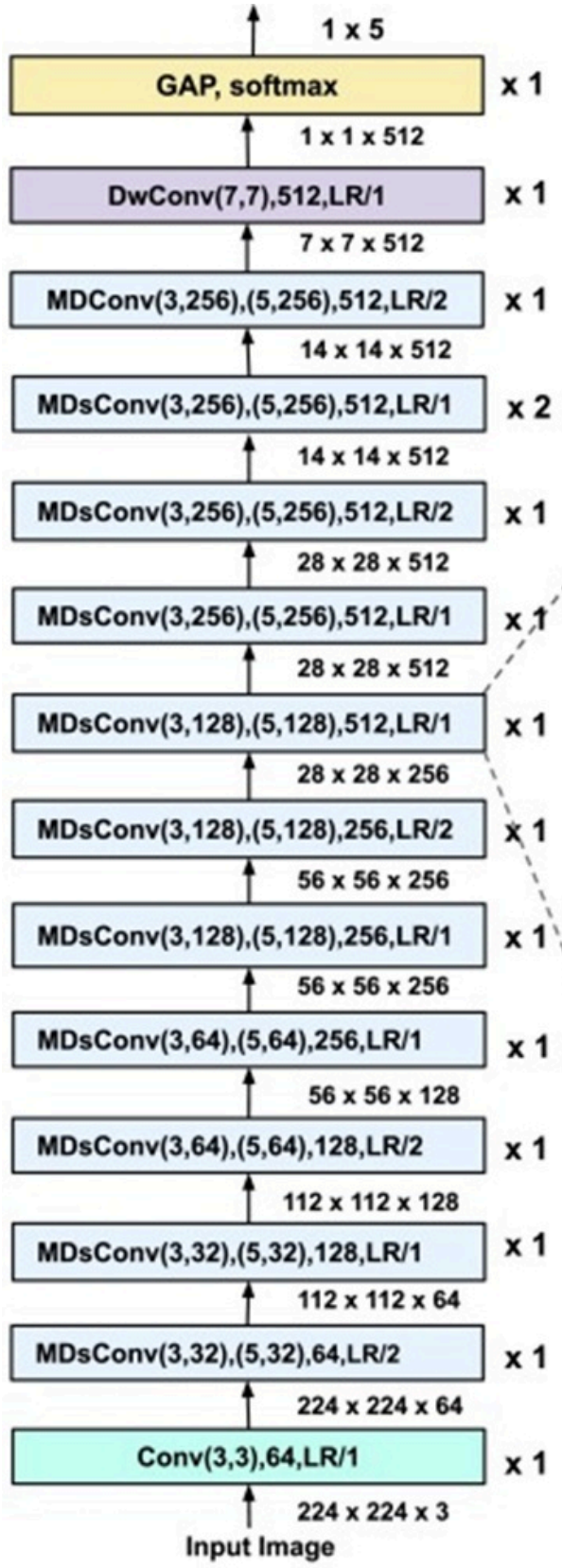


Fig. (a) Proposed Lite-MDC Architecture

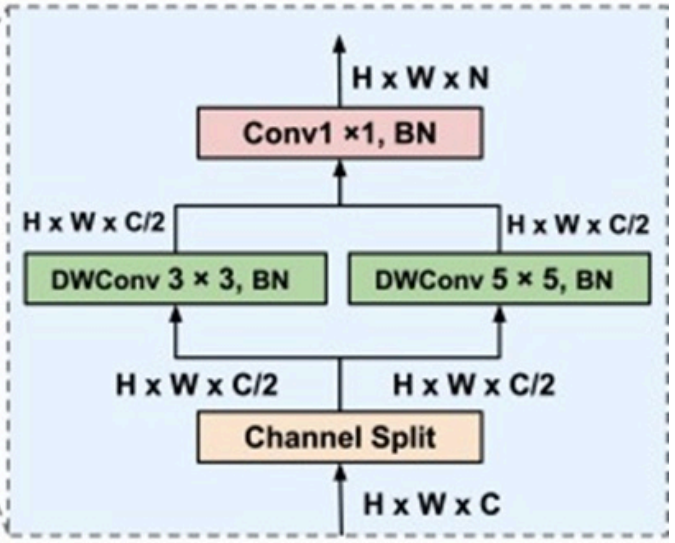


Fig. (b) Proposed MDsConv block

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Figure 8. (a) shows the proposed Lite-MDC architecture, which includes MDsConv blocks referred to as MDsConv[(K1, No. of filters), (K2, No. of filters), No. of filters for pointwise convolution, LR/stride: LeakyReLU / 1 or 2, where K1:3 x 3 kernel, K2:5 x 5 kernel, GAP: Global Average Pooling, and (b) shows the proposed MDsConv block where DWConv: Depthwise convolution, BN: Batch Normalization, and Conv: Standard convolution.

Table 5. Proposed lite-MDC architecture.

Layer name	Output shape	Stride	Filters size
Input image	224 × 224 × 3	-	-
Conv0	224 × 224 × 64	1	3 ×3 × 3× 64
MDConv_1	112 × 112 × 64	2	3 ×3× 32, 5 ×5× 32
PConv_1	112 × 112 × 64	-	1 ×1 × 64× 64
MDConv_2	112 × 112 × 64	1	3 ×3× 32, 5 ×5× 32
PConv_2	112 × 112 × 128	-	1 ×1 × 128× 128
MDConv_3	56 × 56 × 128	2	3 ×3× 64, 5 ×5× 64
PConv_3	56 × 56 × 128	-	1 ×1 × 128× 128
MDConv_4	56 × 56 × 128	1	3 ×3× 64, 5 ×5× 64
PConv_4	56 × 56 × 256	-	1 ×1 × 128× 256
MDConv_5	56 × 56 × 256	1	3 ×3× 128, 5 ×5× 128
PConv_5	56 × 56 × 256	-	1 ×1 × 256× 256
MDConv_6	28 × 28 × 256	2	3 ×3× 128, 5 ×5× 128
PConv_6	28 × 28 × 256	-	1 ×1 × 256× 256
MDConv_7	28 × 28 × 256	1	3 ×3× 128, 5 ×5× 128
PConv_7	28 × 28 × 512	-	1 ×1 × 256× 512
MDConv_8	28 × 28 × 512	1	3 ×3× 256, 5 ×5× 256
PConv_8	28 × 28 × 512	-	1 ×1 × 512× 512
MDConv_9	14 × 14 × 512	2	3 ×3× 256, 5 ×5× 256
PConv_9	14 × 14 × 512	-	1 ×1 × 512× 512
MDConv_10	14 × 14 × 512	1	3 ×3× 256, 5 ×5× 256
PConv_10	14 × 14 × 512	-	1 ×1 × 512× 512
MDConv_11	14 × 14 × 512	1	3 ×3× 256, 5 ×5× 256
PConv_11	14 × 14 × 512	-	1 ×1 × 512× 512
MDConv_12	7 × 7 × 512	2	3 ×3× 256, 5 ×5× 256
PConv_12	7 × 7 × 512	-	1 ×1 × 512× 512
DwConv_13	1 × 1 × 512	1	7 ×7× 512

© 2024 The Author(s). | Published by Agro Intelligence Research Press

PConv_13	1 × 1 × 512	-	1 ×1×512× 512
Conv_14	1 × 1 × 512	1	1 ×1×512× 512
Global average pooling	512		1 × 1
Softmax	5		Classifier

Computational complexity calculations for lite-MDC

Totals for parameters, along with the computations (multiplications combined with additions) required when routing one image through the entire network, are computed as follows: RGB images measuring 224×224×3 feed into the initial convolutional layer, denoted H×W×C. Applied kernels have a 3×3 K×K dimension, along with N=64 kernels. Multiplication sums, addition sums, and parameter requirements for that layer are given in **Eqs. 5-7**.

$$\begin{aligned} Mul_Conv0 &= H \times W \times C \times K \times K \times N \\ &= 224 \times 224 \times 3 \times 3 \times 3 \times 64 \end{aligned} \quad (5)$$

$$\begin{aligned} Add_Conv0 &= H \times W \times (K \times K \times C - 1) \times N \\ &= 224 \times 224 \times (3 \times 3 \times 3 - 1) \times 64 \end{aligned} \quad (6)$$

$$\begin{aligned} Parm_Conv0 &= C \times N + N \\ &= 3 \times 64 + 64 \end{aligned} \quad (7)$$

Following that, multi-kernel depth-wise convolution runs on the (224×224×64) result from the earlier layer (Conv0). The application involves two kernel varieties — K1×K1 at 3×3 and K2×K2 at 5×5 — each appearing N=64 times with a stride of 2. Multiplication, addition, and parameter aggregates for this appear in **Eqs. 8-10**.

$$\begin{aligned} Mul_MDWConv1 &= H \times W \times \frac{C}{2} \times (K_1 \times K_1 + K_2 \times K_2) \\ &= 112 \times 112 \times 32 \times (3 \times 3 + 5 \times 5) \end{aligned} \quad (8)$$

$$\begin{aligned} Add_MDWConv1 &= H \times W \times \frac{C}{2} \\ &\times \{(K_1 \times K_1 - 1) + (K_2 \times K_2 - 1)\} \\ &= 112 \times 112 \times 32 \times \{(3 \times 3 - 1) + (5 \times 5 - 1)\} \end{aligned} \quad (9)$$

$$\begin{aligned} Parm_MDWConv &= (3 \times 3 \times \frac{C}{2} + \frac{C}{2}) + (5 \times 5 \times \frac{C}{2} + \frac{C}{2}) \\ &= (3 \times 3 \times 32 + 32) + (3 \times 3 \times 32 + 32) \end{aligned} \quad (10)$$

Afterward, the (112×112×64) output from MDConv_1 is fed into pointwise convolution. Linearly merging the outputs from the multi-kernel depthwise stage produces an updated feature map. Multiplication, addition, and parameter counts in the

© 2024 The Author(s). | Published by Agro Intelligence Research Press

pointwise operation reside in **Eqs. 11-13**.

$$Mul_PWConv1 = H \times W \times C \times N = 112 \times 112 \times 64 \times 64 \quad (11)$$

$$Add_PWConv1 = H \times W \times (C-1) \times N = 112 \times 112 \times (64-1) \times 64 \quad (12)$$

$$Parm_PWConv1 = C \times N + N = 3 \times 64 + 64 \quad (13)$$

The identical assessment method applies onward to the remaining network layers. Overall, handling one image with the Lite-MDC architecture requires 1167M additions and multiplications, plus 2.2M parameters.

Experimentation

This section discusses the Evaluation Metrics, model training, and implementation details.

Evaluation metrics

Judging how well the suggested model performs relies on metrics such as Precision, Recall, F1-Score, and Accuracy. These derive from calculations in **Eqs. 14-16** in that order. In this context, True positives (TP) indicate classes identified correctly, False Positive (FP) indicates classes wrongly flagged as positive, False Negative (FN) indicates classes erroneously tagged as negative, while True Negative (TN) indicates negative classes recognized properly.

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

$$Recall/Sensitivity = \frac{TP}{TP+FN} \quad (15)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (16)$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (17)$$

5.2. Training and implementation

The research conducts trials across four datasets — a custom-developed pigeon-pea collection and three open datasets: Plant Village [6], Cassava [42], and Apple [43]. Building deep neural networks often requires large amounts of data, which

© 2024 The Author(s). | Published by Agro Intelligence Research Press

can be expanded through data augmentation. Eight distinct alterations affect the images: rotation (range = 360), width shift (range = 0.2), height shift (range = 0.2), shear (range = 0.2), zoom (range = 0.2), brightness (range = 0.1-1.0), horizontal flip, vertical flip, and nearest fill mode. Runtime generation of random values inside the ranges effectively doubles the number of input images. Augmentation improves model training and enhances generalization for classifying pigeon-pea diseases.

Keras is used to implement the Lite-MDC model, which runs on TensorFlow at the core during training [49]. The setup runs Ubuntu 16.04, powered by an Intel 2.54 GHz Processor, equipped with 16 GB of RAM, and supported by an NVIDIA Tesla P100 GPU. Training uses a batch size of 16, the Adam optimizer [50], and categorical cross-entropy loss. Dynamic learning rate tuning activates when validation loss plateaus for 5 consecutive epochs, triggering a 0.3 reduction in the learning rate. Full Lite-MDC training spans all four datasets using matching hyperparameters and conditions to maintain balanced assessment. Division of each dataset allocates 60% for training, 20% for validation, and 20% for testing. Image inputs stay standardized at 224×224×3. Once trained on the training portion, model performance is evaluated using Precision, Recall, F1-Score, and Accuracy on the held-out test data.

Results and Discussion

This section covers the outcomes and insights from the suggested technique, including testing on the newly created dataset and three other openly accessible datasets. Evaluations contrast the findings with current top-performing approaches, using both qualitative and quantitative lenses, to provide a comprehensive overview. A compact CNN model incorporating multi-kernel depthwise separable convolution, termed Lite-MDC, was developed for plant disease identification. Depthwise convolution reduces processing load, while varied convolution kernels align receptive fields with the multi-scale patterns observed in plant diseases. An in-depth contrast of the introduced MDsConv against standard convolution appears next:

Comparison with standard convolution

Direct benchmarking of MDsConv versus StdConv examines their processing requirements and parameter requirements during a single convolution step. The cost reflects the combined total of the multiplication and addition steps. Take an input feature map sized 224×224×16 having input channels (C) = 16, Stride = 1, Padding = same, and Number of filters (N) = 64. Standard convolution at Kernel size K = 3×3 results in: Parameter = 9280, Cost = 921.6 M. With matching conditions, multi-kernel depthwise convolution using kernel size K1=3×3 along with two kernels K2=5×5 gives: Parameter = 1376, Cost = 14.20 M. MDsConv features seven times fewer parameters and 65 times reduced computational cost relative to standard convolution, with full specifics in **Table 6**. Inference duration is reduced thanks to lower computation and parameter counts, enabling faster execution. The developed architecture achieves leading performance in plant disease categorization while keeping computational demands and memory usage minimal. Complete computational complexity contrasts are detailed below:

Table 6. Comparison of parameters and cost calculation for single operation in StdConv and MDsConv, considering Input feature map shape = (224 × 224 × 16), Stride = 1, Padding = 0, Output channels = 64. StdConv uses Kernel K = 3 × 3 and MDsConv uses two kernels K1 = 3 × 3, K2 = 5 × 5.

Method	Input	Kernels	Mul_Add	Parameter
StdConv	N = 64, C = 16	3 × 3	921.6 M	9,280
MDsConv	N = 64, C = 16	3×3**,** 5×5	**14.20 M**	**1,376**

Computational complexity comparison

© 2024 The Author(s). | Published by Agro Intelligence Research Press

As outlined earlier in section 4.4, **Eqs. 5-13** compute the full computations and parameters needed by the Lite-MDC network. Overall, this yields 1167 M (additions and multiplications) along with 2.2 M parameters to process one image. These values remain far below those demanded by a typical VGG16 network. The VGG16 architecture requires 15,300 M of computations and 140 M of parameters. Parameter and computation contrasts between Lite-MDC and standard VGG16 are presented in **Table 7**. Lite-MDC clearly delivers notable accuracy improvements with fewer parameters and computations, thereby reducing inference time and improving speed. Therefore, the Lite-MDC design is well-suited to real-time scenarios and outperforms state-of-the-art techniques on the Pigeon-pea, Plant village, Cassava, and Apple datasets. Experimental outcomes feature in **Tables 8 and 9**.

Table 7. Comparison of Lite-MDC and VGG16 in terms of parameters and computations.

Method	Total parameter	Accuracy	Computations
VGG16 [9]	140 M	87.33%	15,300 M
Lite-MDC (Proposed)	**2.2 M**	**95.34%**	**1,167 M**

Table 8. Comparison of the proposed method, Lite-MDC, on different datasets.

Dataset	Number of Parameters	Accuracy (%)	F1-Score	Recall	Precision
Cassava Dataset [42]	2.2 M	87.20%	91	90	86
Apple Foliar Disease Dataset [43]	2.2 M	97.36%	91	90	86
PlantVillage Dataset [6]	2.2 M	99.77%	99	99	99
Pigeon-pea Dataset (proposed)	2.2 M	95.01%	95	94	94

Table 9. Class-wise accuracy using the proposed method architecture on the Pigeon-pea dataset, Cassava, and Apple leaf dataset.

Dataset	Class	Accuracy (%)	F1-Score	Recall	Precision	Incorrect predictions	Correct predictions	Total samples
Pigeon-pea (Proposed)	Dry root rot (DRR)	98.0	97	99	95	3	217	220
	Healthy	90.0	91	90	92	25	227	252
	Phytophthora blight (PB)	92.0	93.0	92	95	27	313	340
	Sterility mosaic (SM)	97.0	97.0	97	96	21	709	730
	Wilt disease (WD)	93.7	95	95	95	28	417	445
Cassava Leaf Dataset	Cassava Bacterial Blight (CBB)	50.0	52	50	56	33	76	109
	Cassava Brown Streak Disease (CBSD)	78.2	80	78	83	48	171	219
	Cassava Green	77.0	80	77	84	27	183	210

© 2024 The Author(s). | Published by Agro Intelligence Research Press

	Mottle (CGM)							
	Cassava Mosaic Disease (CMD)	96.0	95	96	95	21	1259	1280
	Healthy	93.7	95	95	95	28	188	216
Apple Leaf Dataset	Healthy leaves	99.00	99	99	98	1	102	103
	Multiple diseases	61	67	56	83	7	11	18
	Apple Rust	99.0	98	99	96	97	124	221
	Apple Scab	98.3	98	98	97	2	117	119

Comparison with state-of-the-art methods on the pigeon-pea dataset

The Lite-MDC architecture is benchmarked against advanced networks, including ResNet, Xception, InceptionV3, and DenseNet121, on the pigeon-pea dataset. Comparisons further include compact leading architectures such as MobileNet, MobileNetV2, MobileNetV3, and NASNet Mobile. Relevant metrics on accuracy, parameters, frames per second (FPS), and floating-point operations (FLOPS) are listed in **Table 10**. Lite-MDC surpasses previous techniques, securing 94.14% accuracy. It relies on just 2.2 million parameters — the lowest among those shown in **Table 10** and 63 times smaller than standard VGG-16. A plot of accuracy against parameter counts for Lite-MDC and other leading techniques is shown in **Figure 9**. The reduced parameter count in Lite-MDC arises from its multi-kernel depth-wise convolution. Varied kernels additionally support the extraction of multi-scale plant disease traits. Findings indicate high accuracy, short inference times, and peak FPS. This confirms Lite-MDC's capability and fitness for live plant disease spotting. Class-specific metric evaluations on the pigeon pea dataset are shown in **Table 9**. The confusion matrix, receiver operating characteristic (ROC), and area under the curve (AUC) are illustrated in **Figures 10a and 10b**.

Table 10. Comparison with state-of-the-art methods for pigeon-pea disease detection.

Model	FLOPS	Frames per second (FPS)	Accuracy	Approximate no. of parameters
Standard VGG16	31.1 G	5	87.33	140 M
ResNet50	7.85 G	7	79.34	25,63 M
InceptionV3	6.10 G	9	89.56	23.85 M
Xception	9.13 G	10	88.30	22.91 M
DenseNet121	7.04 G	8	86.37	7.0 M
EfficientNetB0	0.89 G	23	92.30	5.33 M
NASNet mobile	1.15 G	23	84.90	5.3 M
MobileNet	1.15 G	26	83.89	4.2 M
MobileNetV2	0.93 G	28	86.63	3.5 M
Lite-MDC (Proposed)		**33**	**94.14**	**2.2 M**
	0.61 G			

© 2024 The Author(s). | Published by Agro Intelligence Research Press

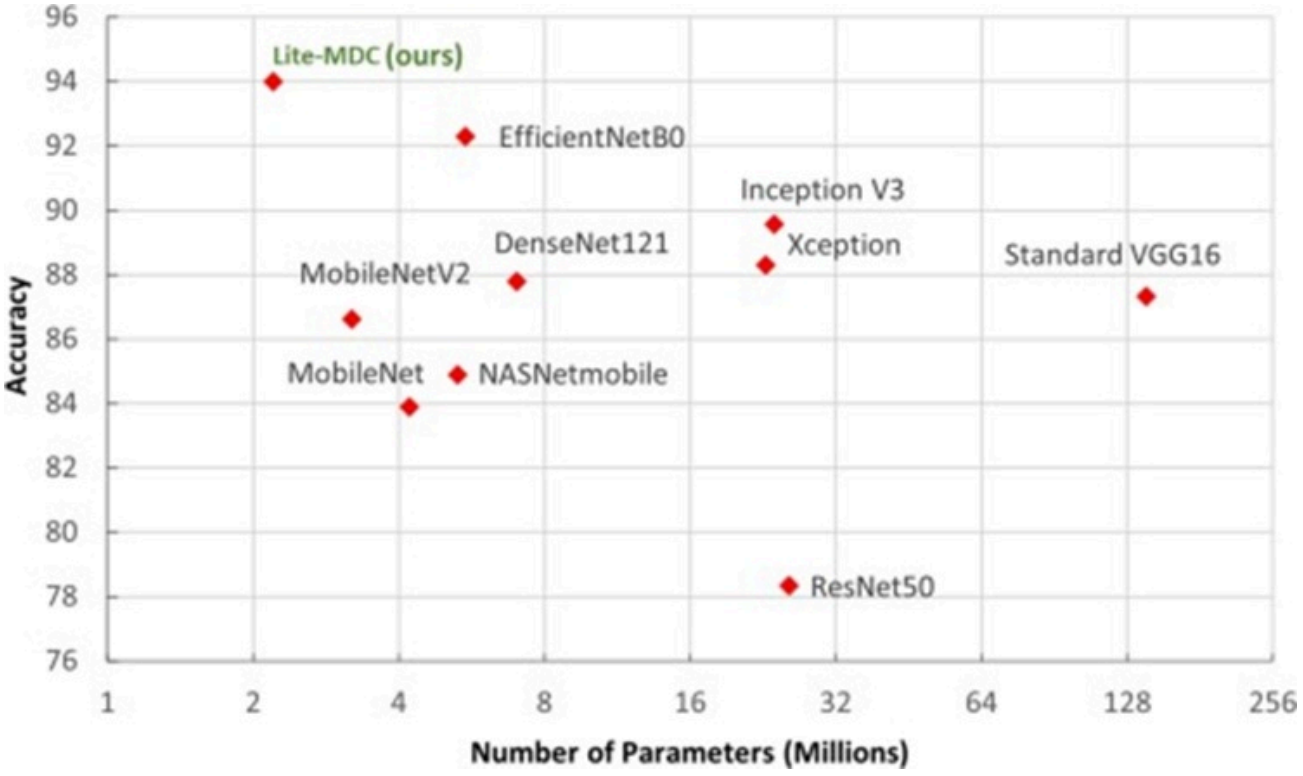
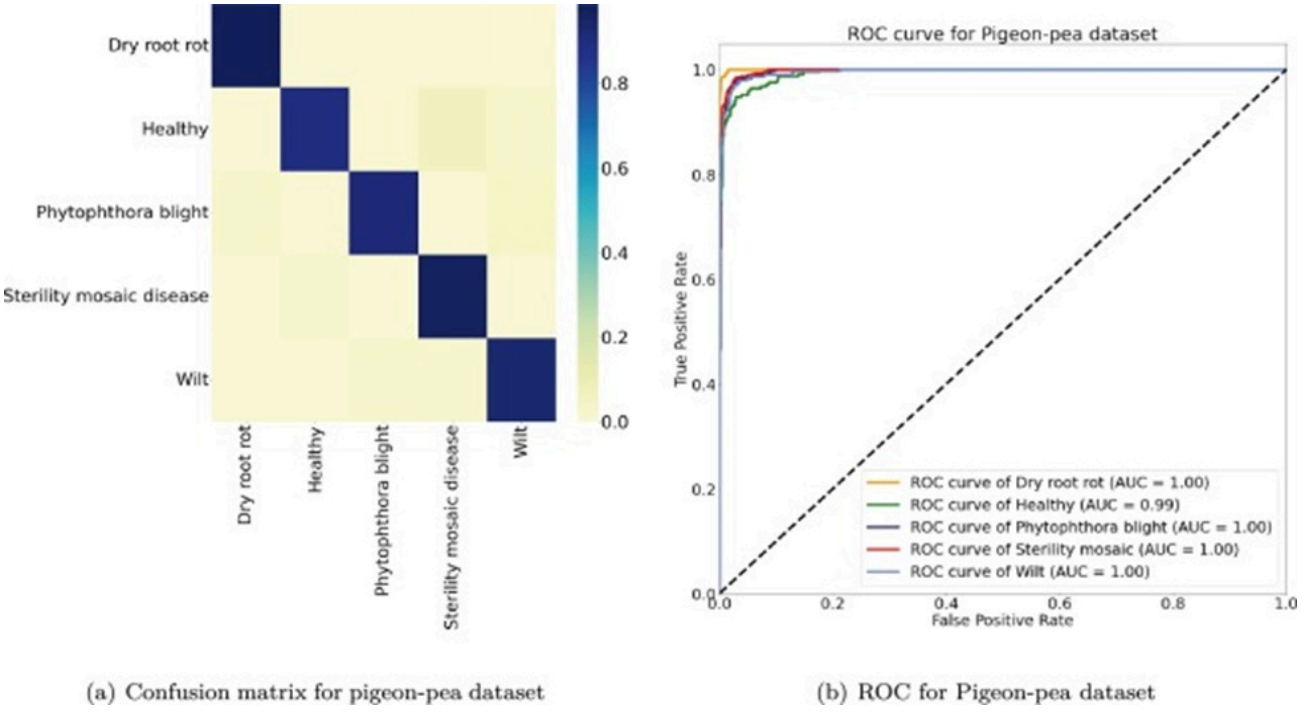


Figure 9. Test Accuracy vs Number of Parameters on Pigeon-pea dataset, where the x-axis shows the number of parameters in log scale and the y-axis shows Test accuracy on pigeon-pea dataset.



(a) Confusion matrix for pigeon-pea dataset (b) ROC for Pigeon-pea dataset

Figure 10. Proposed Lite-MDC (a) confusion matrix and (b) ROC Curve for the pigeon-pea dataset.

Comparison with existing studies on public datasets

© 2024 The Author(s). | Published by Agro Intelligence Research Press

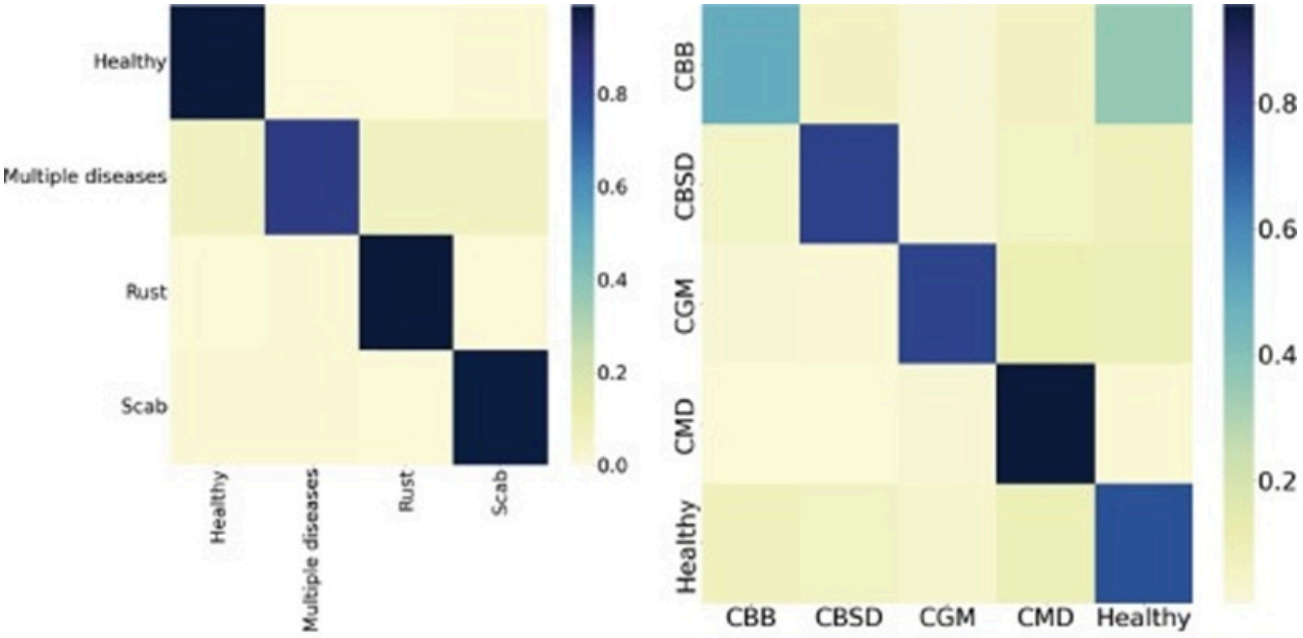
Assessment of the technique took place on three open datasets: (1) Plant Village dataset, (2) Cassava dataset, (3) Apple dataset. The contrasts center on total parameters and accuracy rates shown in **Table 11**. The new architecture exceeds earlier works across all datasets. Nearly all methods achieve around 99% on the PlantVillage dataset **(Table 11)**, since its images originate from indoor environments with fixed backgrounds and lack multi-scale samples, as seen in **Figure 6(c)**, reducing their applicability to real-world situations. To test its true outdoor effectiveness, the model was also evaluated on the Cassava and Apple leaf datasets. Cassava images exhibit multi-scale disease variations. According to **Table 11**, the Standard ResNet50 [14] model achieves 81.5% accuracy with 25.64 M parameters, whereas Lite-MDC reaches 86.4% accuracy with 2.2 M parameters. Total parameters shrink by 11x compared to the ResNet50 structure. The Apple leaf dataset features orchard photos. Vo *et al.* [51] obtained 91.54% accuracy using 4.2 M parameters, whereas Lite-MDC achieved 97.2% accuracy with 2.2 M parameters. These findings highlight the technique's strength for live disease identification. Class-level performance breakdowns of the technique on Cassava and Apple leaf datasets are given in **Table 9**. The final four columns display assessment indicators: Precision, Recall, F1-Score, and Accuracy. The table additionally lists per-class sample counts along with correct and incorrect predictions. Confusion matrices for the apple and cassava datasets are shown in **Figures 11a and 11b**. Receiver operating characteristic (ROC) curves and area under the curve (AUC) for the apple and cassava datasets are shown in **Figures 12a and 12b**, respectively. **Figure 9** shows the leading method's performance across the three datasets.

Table 11. Comparison of the proposed Lite-MDC with state-of-the-art methods from the literature on plan village, cassava, and apple leaf datasets.

Dataset	Method	Accuracy %	Parameter
PlantVillage [6]	AlexNet [38]	99.24%	126 M
	DenseNet169 [16]	99.89%	14.31 M
	GoogleNet [15]	99.76%	23.85 M
	ResNet34 [14]	99.67%	21.6 M
	ResNet152 [14]	99.69%	60.42 M
	VGG13 [9]	99.49%	135.6 M
	Squeezenet [37]	97.20%	1.25 M
	Liu *et al.* [47]	99.78%	-
	Lite-MDC (Proposed)	**99.78%**	**2.2 M**
Cassava Leaf Dataset [42]	VGG16 [9]	70.1%	140 M
	MobileNet [12]	68.5%	4.25 M
	ResNet50 [14]	81.5%	25.64 M
	Ayu *et al.* [52]	65.06%	3.53 M
	Alabdulmohsin *et al.* [53]	68.8%	-
	Lite-MDC (Proposed)	**86.4%**	**2.2 M**
Apple Leaf Dataset [43]	AlexNet [38]	65.6%	62.37 M
	VGG16 [9]	90.1%	140 M
	MobileNetV2 [12]	85.98%	3.5 M
	ResNetV2 [14]	93.68%	44 M

© 2024 The Author(s). | Published by Agro Intelligence Research Press

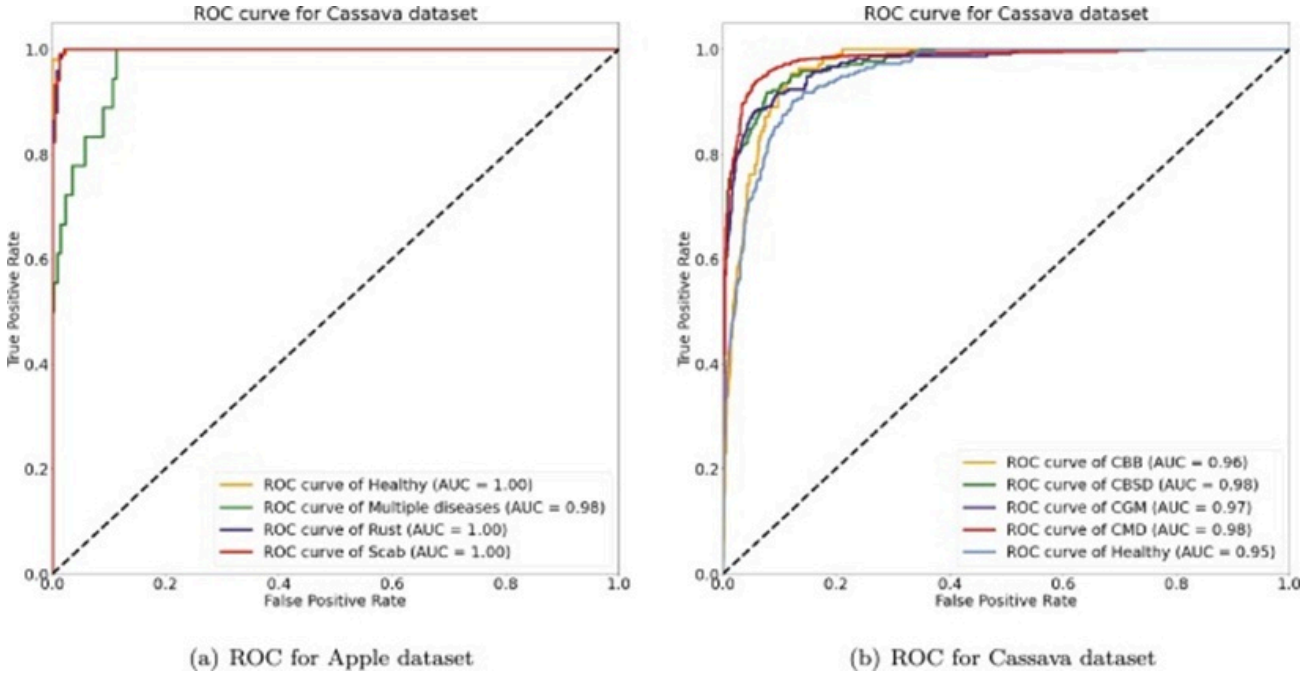
InceptionV3 [15]	89.28%	23.85 M
Manaswini *et al.* [54]	73.56%	-
Vo *et al.* [51]	91.54%	4.2 M
Alsayed *et al.* [55]	68.8%	-
Lite-MDC (Proposed)	**97.2%**	**2.2 M**



(a) Apple dataset (b) Cassava dataset

Figure 11. Confusion matrix for Apple and Cassava datasets in figure (a) Apple dataset with four classes (b) cassava diseases as Cassava Bacterial Blight (CBB), Cassava Brown Streak Disease (CBSD), Cassava Green Mottle (CGM), Cassava Mosaic Disease (CMD), and Healthy.

© 2024 The Author(s). | Published by Agro Intelligence Research Press



(a) ROC for Apple dataset (b) ROC for Cassava dataset

Figure 12. ROC-AUC curve for the Apple and cassava dataset as shown in Figure (a) and Figure (b), respectively.

Visualization

Gradient-weighted class activation heatmaps help show how different techniques focus attention on diseased regions [56], taken from the pigeon-pea dataset, as presented in **Figure 13**. Every row stands for one disease: (I) Sterility mosaic, (II) Healthy, (III) Wilt, (IV) Dry root rot, (V) Phytophthora blight. Columns correspond to various models: Proposed Lite-MDC, DenseNet121, ResNet50, and MobileNetV1 in (a), (b), (c), and (d). Visuals make it evident that Lite-MDC concentrates more effectively on affected zones than typical leading techniques. DenseNet121, ResNet50, and MobileNetV1 tend to distribute attention unevenly across disease areas, missing important disease indicators. Hence, the multi-kernel method successfully gathers multi-scale features and accurately targets meaningful regions.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

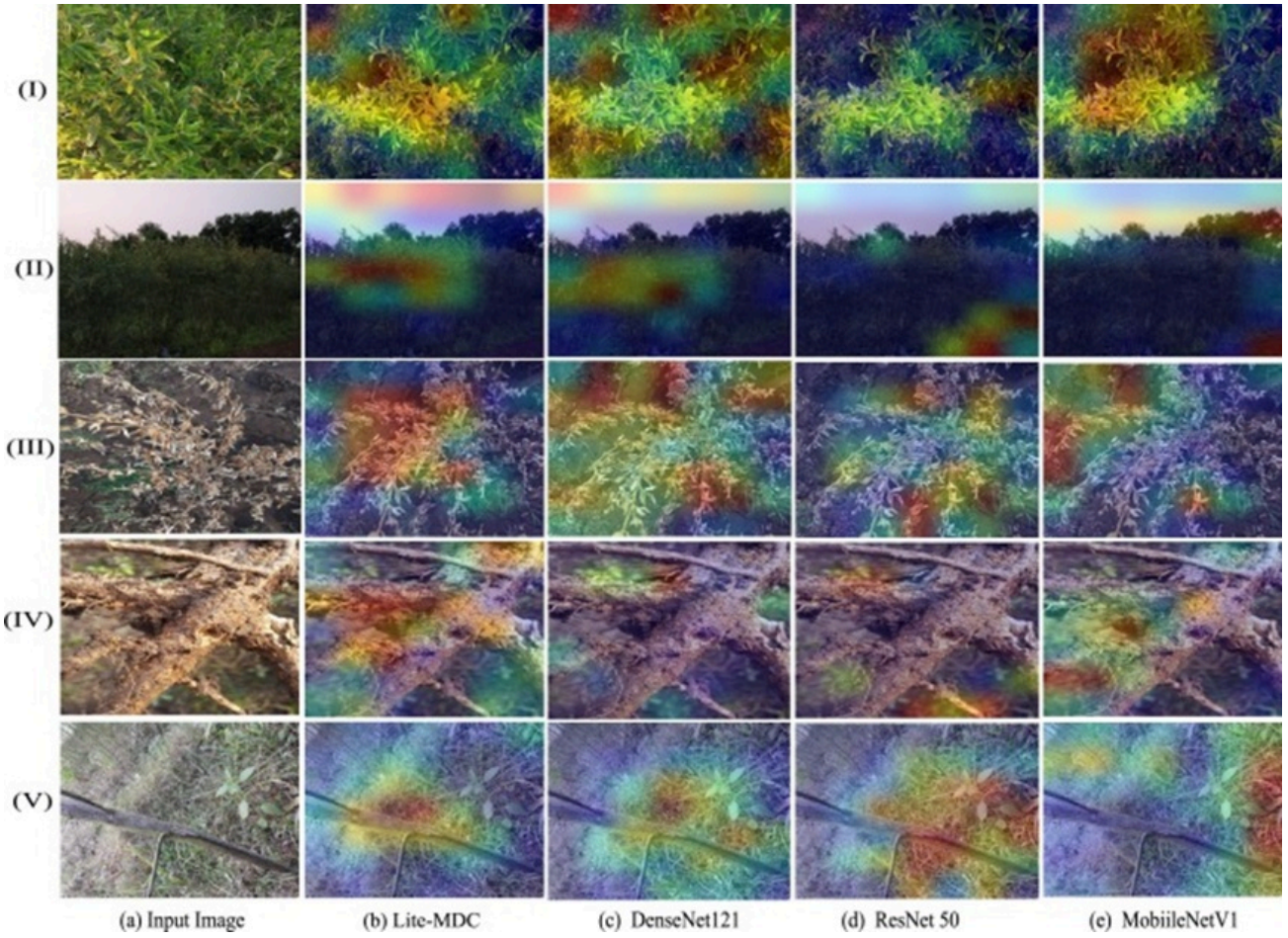


Figure 13. Qualitative results GARD-CAM visualization comparison with other methods, row indicates disease classes and column indicates models (a) Input image, the feature map from the last convolution layer of (b) Lite-MDC, (c) DenseNet121, (d) ResNet50, (e) MobileNet and diseases as (I) sterility mosaic, (II) healthy (III) wilt, (IV) dry root rot, and (V) phytophthora blight.

Ablation study

Efforts center on building a compact structure for live plant disease identification. Standard VGG16 served as the reference due to its excellent results on the ImageNet benchmark for image categorization. The main drawback of the standard VGG16 is its high computational cost and parameter count. To solve this, resource-intensive layers were pinpointed and optimized in stages: First, fully connected layers in VGG16 were identified as costly and replaced with depthwise convolutions without sacrificing performance, called Modification-1. Next, ordinary convolutions were replaced with depthwise convolutions to reduce the number of parameters further while maintaining performance, resulting in Modification-2. Experiments showed that depthwise convolutions with a single 3×3 kernel cannot capture multi-scale plant disease traits and achieve lower accuracy. In the end, depthwise convolution was replaced with multi-kernel depthwise separable convolution (MDsConv), with 3×3 and 5×5 kernels to address multi-scale challenges, labeled Modification-3. Dropout was also added in the initial Lite-MDC layers with climbing rates (0.1 to 0.5) to fight overfitting, designated Modification-4. Lite-MDC beats current leading techniques by hitting 95% accuracy using 2.2 M parameters as listed in **Table 12**.

Table 12. Ablation study of Lite-MDC on Pigeon-pea Dataset (DWConv: Depthwise convolution, MDsConv: Multi-kernel depthwise separable Convolution, FC: Fully Connected layer).

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Methods	(DWConv in FC)	DWConv	MDsConv	Dropout	FLOPS	Parameter	Accuracy
VGG-16					31.0 G	140 M	87.33
Lite-MDC-Mod1	✓				10.70 G	15 M	90.53
Lite-MDC-Mod2	✓	✓			0.60 G	2.2 M	91.58
Lite-MDC-Mod3	✓		✓		0.61 G	2.21 M	93.80
Lite-MDC (Proposed)	✓		✓	✓	**0.61 G**	**2.21 M**	**94.14**

The Lite-MDC design works well on resource-limited platforms such as agricultural robots and mobile phones. Mobile devices provide handy support for farmers and specialists when identifying pigeon-pea diseases. Scouting robots are widely used in crop monitoring to refine farming practices. In addition, the architecture can be integrated into scouting robots for disease spotting and recognition. Yet, as a classification system, it cannot specify precise disease locations within images—a key drawback of the current method. This issue can be resolved later using object detection models supported by bounding box labels.

Conclusion

Presented here is the initial comprehensive pigeon-pea dataset assembled from eighteen sites for plant disease identification, paired with the Lite-MDC technique for efficient processing. This plant disease identification model uses 2.2M parameters. The architecture provides strong accuracy, efficient real-time processing, and low memory requirements. Multi-kernel depthwise convolution using 3×3 and 5×5 sizes enhances multi-scale feature capture for plant disease identification. Depthwise convolution reduces computational complexity and the number of network parameters. The technique needs sixty-three times fewer parameters than the VGG16 architecture. Moreover, comparisons with InceptionV3, VGG16, ResNet50, DenseNet, MobileNet, MobileNetV3, NASNet, SqueezeNet, and the new method confirm advantages in accuracy and speed, achieving 94.14% accuracy and 33 frames per second (FPS) on an NVIDIA P100 GPU with 16 GB RAM. Testing was performed on four datasets: the new pigeon-pea collection, the Plant Village dataset, the Cassava dataset, and the apple leaf dataset, achieving 94.14%, 99.78%, 86.4%, and 97.2% accuracy, respectively. Future efforts will examine real-time behavior on constrained platforms such as smartphones and agricultural robots to support practical plant disease detection.

Acknowledgements

The work is funded by the MHRD, India, and is supported by Research and development under NDF, All India Council for Technical Education (AICTE), New Delhi. We extend our sincere gratitude to the scientists from the agricultural research stations and the Indian Institute of Pulse Research centers for their contributions to the novel pigeon pea data collection.

Conflict of interest

None

Financial support

None

Ethics statement

None

© 2024 The Author(s). | Published by Agro Intelligence Research Press

Received: 02 Jun 2023 Revised: 07 Aug 2023 Accepted: 24 Aug 2023
Published online: 25 November 2024

Rights and permissions

Open Access The author(s) retain copyright. This article is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. It may be shared and adapted for non-commercial purposes with appropriate attribution, an indication of changes, and distribution of adaptations under the same license. Third-party material may be subject to separate terms identified in its credit line. View the license at https://creativecommons.org/licenses/by-nc-sa/4.0/.

References

1. Sarkar S, Panda S, Yadav K, Kandasamy P. Pigeon pea (Cajanus cajan) an important food legume in Indian scenario – a review. Legume Res. 2020;43(5):601-10.

2. Savary S, Ficke A, Aubertot JN, Hollier C. Crop losses due to diseases and their implications for global food production losses and food security. Food Secur. 2012;4(4):519-37.

3. Strange RN, Scott PR. Plant disease: A threat to global food security. Annu Rev Phytopathol. 2005;43:83-116.

4. Miller SA, Beed FD, Harmon CL. Plant disease diagnostic capabilities and networks. Annu Rev Phytopathol. 2009;47:15-38.

5. Mohanty SP, Hughes DP, Salathé M. Using deep learning for image-based plant disease detection. Front Plant Sci. 2016;7:1419.

6. Hughes DP, Salathé M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060. 2015.

7. Agarwal M, Gupta SK, Biswas K. Development of efficient CNN model for tomato crop disease identification. Sustain Comput Inform Syst. 2020;28:100407.

8. Li L, Zhang S, Wang B. Plant disease detection and classification by deep learning—a review. IEEE Access. 2021;9:56683-98.

9. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014.

10. Chollet F. Xception: Deep learning with depthwise separable convolutions. Proc IEEE Conf Comput Vis Pattern Recognit. 2017.

11. Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. Proc IEEE Conf Comput Vis Pattern Recognit. 2018:8697-710.

12. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017.

13. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: Inverted residuals and linear bottlenecks. Proc IEEE Conf Comput Vis Pattern Recognit. 2018:4510-20.

14. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc IEEE Conf Comput Vis Pattern Recognit. 2016:770-8.
https://doi.org/10.1109/CVPR.2016.90.

15. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. Proc IEEE Conf Comput Vis Pattern Recognit. 2016:2818-26.

16. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. Proc IEEE Conf Comput Vis Pattern Recognit. 2017:4700-8.

17. Tan M, Le Q. EfficientNet: Rethinking model scaling for convolutional neural networks. Proc Int Conf Mach Learn. 2019:6105-14.

18. Sankaran S, Ehsani R. Comparison of visible-near infrared and mid-infrared spectroscopy for classification of huanglongbing and citrus canker infected leaves. Agric Eng Int CIGR J. 2013;15(3):75-9.

19. Revathi P, Hemalatha M. Identification of cotton diseases based on cross information gain deep forward neural network classifier with pso feature selection. Int J Eng Technol. 2014;5(6):4637-42.

20. Karmokar BC, Ullah MS, Siddiquee MK, Alam KMR. Tea leaf diseases recognition using neural network ensemble. Int J Comput Appl. 2015;114(17).

21. Chaudhary P, Chaudhari AK, Cheeran A, Godara S. Color transform based approach for disease spot detection on plant leaf. Int J Comput Sci Telecommun. 2012;3(6):65-70.

22. Patil SB, Bodhe SK. Leaf disease severity measurement using image processing. Int J Eng Technol. 2011;3(5):297-301.

23. Patil JK, Kumar R. Feature extraction of diseased leaf images. J Signal Image Process. 2012;3(1):60-3.

24. Reed TR, Dubuf JH. A review of recent texture segmentation and feature extraction techniques. CVGIP Image Underst. 1993;57(3):359-72.

25. Hlaing CS, Zaw SMM. Tomato plant diseases classification using statistical texture feature and color feature. 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). IEEE. 2018:439-44.

26. Al-Hiary H, Bani-Ahmad S, Reyalat M, Braik M, Alrahamneh Z. Fast and accurate detection and classification of plant diseases. Int J Comput Appl. 2011;17(1):31-8.

27. Waghmare H, Kokare R, Dandawate Y. Detection and classification of diseases of grape plant using opposite colour local binary pattern feature and machine learning for automated decision support system. 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN). IEEE. 2016:513-8.

28. Harakannanavar SS, Rudagi JM, Puranikmath VI, Siddiqua A, Pramodhini R. Plant leaf disease detection using computer vision and machine learning algorithms. Glob Transit Proc. 2022;3(1):305-10.

29. Kawasaki Y, Uga H, Kagiwada S, Iyatomi H. Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. Int Symp Vis Comput. Springer. 2015:638-45.

30. 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. 2016:2016.

31. Zhang K, Wu Q, Liu A, Meng X. Can deep learning identify tomato leaf disease? Adv Multimed. 2018:2018.

32. Selvaraj MG, Vergara A, Ruiz H, Safari N, Elayabalan S, Ocimati W, et al. AI-powered banana diseases and pest detection. Plant Methods. 2019;15(1):1-11.

33. Liu J, Wang X. Tomato diseases and pests detection based on improved yolo v3 convolutional neural network. Front Plant Sci. 2020;11:898.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

34. Wang G, Sun Y, Wang J. Automatic image-based plant disease severity estimation using deep learning. Comput Intell Neurosci. 2017:2017.

35. Biswas M, Ray M, Jahan I, Khan S, Ahmad Saad S, Bharman P. Deep learning in agriculture: A review. Asian J Res Comput Sci. 2022:28-47.

36. Lu Y, Yi S, Zeng N, Liu Y, Zhang Y. Identification of rice diseases using deep convolutional neural networks. Neurocomputing. 2017;267:378-84.

37. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. Squeezenet: AlexNet-level accuracy with 50x fewer parameters and <0.5 mb model size. arXiv preprint arXiv:1602.07360. 2016.

38. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst. 2012;25:1097-105.

39. Deeba K, Amutha B. Resnet-deep neural network architecture for leaf disease classification. Microprocess Microsyst. 2020:103364.

40. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. Proc IEEE Conf Comput Vis Pattern Recognit. 2015:1-9.

41. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, Inception-ResNet and the impact of residual connections on learning. Proc AAAI Conf Artif Intell. 2017.

42. Mwebaze E, Gebru T, Frome A, Nsumba S, Tusubira J. Icassava 2019 fine-grained visual categorization challenge. arXiv preprint arXiv:1908.02900. 2019.

43. Thapa R, Snavely N, Belongie S, Khan A. The Plant Pathology 2020 challenge dataset to classify foliar disease of apples. arXiv preprint arXiv:2004.11958. 2020.

44. Ramcharan A, Baranowski K, McCloskey P, Ahmed B, Legg J, Hughes DP. Deep learning for image-based cassava disease detection. Front Plant Sci. 2017;8:1852.

45. Yu HJ, Son CH, Lee DH. Apple leaf disease identification through region-of-interest-aware deep convolutional neural network. J Imaging Sci Technol. 2020;64(2):20507.

46. Xie C, Wang R, Zhang J, Chen P, Dong W, Li R, et al. Multi-level learning features for automatic classification of field crop pests. Comput Electron Agric. 2018;152:233-41.

47. 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;30:2003-15.

48. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929-58.

49. Dillon JV, Langmore I, Tran D, Brevdo E, Vasudevan S, Moore D, et al. TensorFlow distributions. arXiv preprint arXiv:1711.10604. 2017.

50. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.

51. Vo HT, Yu GH, Nguyen HT, Lee JH, Dang TV, Kim JY. A study on applying homomorphic filter and Deep Neural Network for apple trees diseases classification. Proc 9th Int Conf Smart Media Appl. 2020:92-6.

52. Ayu H, Surtono A, Apriyanto D. Deep learning for detection cassava leaf disease. J Phys Conf Ser. 2021;1751:012072.

© 2024 The Author(s). | Published by Agro Intelligence Research Press

53. Alabdulmohsin I, Maennel H, Keysers D. The impact of reinitialization on generalization in convolutional neural networks. arXiv preprint arXiv:2109.00267. 2021.

54. Manaswini P, Bhatt JS. Towards glass-box CNNs. arXiv preprint arXiv:2101.10443. 2021.

55. Alsayed A, Alsabei A, Arif M. Classification of apple tree leaves diseases using deep learning methods. Int J Comput Sci Netw Secur. 2021;21(7):324-30.

56. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D. Grad-CAM: visual explanations from deep networks via gradient-based localization. Proc IEEE Int Conf Comput Vis. 2017:618-26.

© 2024 The Author(s). | Published by Agro Intelligence Research Press