Note: Plant Leaf Disease Network (PLeaD-Net): Identifying Plant Leaf Diseases through Leveraging Limited-Resource Deep Convolutional Neural Network

Joyanta Jyoti Mondal*
Md. Farhadul Islam*
joyanta.jyoti.mondal@g.bracu.ac.bd
md.farhadul.islam@g.bracu.ac.bd
BRAC Universty
Dhaka, Bangladesh

Sarah Zabeen
BRAC Universty
Dhaka, Bangladesh
sarah.zabeen@g.bracu.ac.bd

A. B. M. Alim Al Islam
Bangladesh University of Engineering and Technology
Dhaka, Bangladesh
alim_razi@cse.buet.ac.bd

Jannatun Noor
BRAC Universty
Dhaka, Bangladesh
jannatun.noor@bracu.ac.bd

ABSTRACT

Agriculture is the fundamental source of revenue and Gross Domestic Product (GDP) in many countries where economically developing countries; especially the Global South are no exception. Various types of plant-based diseases are strongly intertwined with the everyday lives of those who are connected with agriculture. Among the diseases, most of them can be diagnosed by leaves. However, due to the variety of illnesses, identifying and classifying any plant leaf disease is difficult and time-consuming. Besides, late identifications of diseases cause losses for the farmers on a large scale, which in turn affects their financial state. Therefore, to overcome this problem, we present a lightweight approach (called PLeaD-Net) to accurately recognize and categorize plant leaf diseases in this paper. Here, leveraging a limited-resource deep convolutional network (Deep CNN) model, we extract information from sick sections of a leaf to accurately identify locations of disease. In comparison to existing deep learning methods and other prior research, our proposed approach achieves a much higher performance using fewer parameters as per our experimental results. In our study and experimentation, we develop and implement an architecture based on Deep CNN. We test our architecture on a publicly available dataset that contains different types of plant leaves images and backgrounds.

CCS CONCEPTS

• **Computing methodologies → Neural networks**.

*Both authors contributed equally to this research. And, Corresponding Authors.

COMPASS '22, June 29-July 1, 2022, Seattle, WA, USA

ACM ISBN 978-1-4503-9347-8/22/06…$15.00
https://doi.org/10.1145/3530190.3534844

KEYWORDS

Convolutional Neural Network, Deep Learning, Plant Disease Classification, Leaf Disease Identification, Transfer Learning, Machine Learning

ACM Reference Format:
Joyanta Jyoti Mondal, Md. Farhadul Islam, Sarah Zabeen, A. B. M. Alim Al Islam, and Jannatun Noor. 2022. Note: Plant Leaf Disease Network (PLeaD-Net): Identifying Plant Leaf Diseases through Leveraging Limited-Resource Deep Convolutional Neural Network. In *ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS) (COMPASS '22), June 29-July 1, 2022, Seattle, WA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3530190.3534844

1 INTRODUCTION

Numerous nations that are economically developing [14] depend heavily on agriculture for survival. In this era of modern technology, people who are connected with agriculture are keenly interested to have a touch of technology in their work and adopt modern agricultural approaches. However, they have less access to latest technologies as the cost of contemporary technologies is not generally that much affordable.

Among the latest key technologies, usages of Machine Learning (ML), image processing, and Deep Learning (DL) have increased in recent years. Therefore, techniques for correctly detecting plant disease leveraging these technologies demand more exposure than ever before. When it comes to detecting plant diseases, the initial instinct is to search for visible symptoms of the plant. As a consequence, for the learning model to be capable of recognizing the unique symptoms of each disease, it must also possess exceptional feature extraction and processing abilities. These requirements, in turn, exhibit a demand for heavy computational power to cover a variety of case scenarios. We can find evidences of this happening, as high-end DL models have been utilized to address the complexity and efficacy of precision agricultural research in recent years [2, 15].

On the other hand, farmers in developing countries prefer to adopt conventional methods as the technology is less affordable and less accessible to them. However, conventional approaches (or the manual processes) demand more labor and usually take longer time periods to react in case of having a plant disease. Moreover, it is more likely that manual inspecting often results in ineffective findings. Besides, many farmers have also been found to use pesticides to reduce the effects of illness without first determining the exact disease. In this regard, farmers use pesticides on a regular basis, which has the potential to harm plant quality as well as human health. Here, in the process of identifying plant diseases and using pesticides accordingly, an affordable (lightweight) DL-based plant disease detection and classification can help the farmers more efficiently and promptly.

Accordingly, in this study, we attempt to classify plant leaf diseases via leveraging a comparatively lightweight Deep Convolutional Neural Network (Deep CNN) based architecture that will be generally good on any kind of plant leaf disease images. Here, we adopt the notion of Deep CNNs, as they have already shown promise in a number of real-world applications including image classification, object identification, etc. Based on our study, we make the following set of contributions in the research.

- We propose a new lightweight Deep CNN based architecture to identify plant leaf disease through focusing on a less number of parameters. The consideration of having a less number of parameters enables operations using less computational power [3] while having higher accuracy without using any transfer learning approach.
- We perform experiments in which we implement our model and compare it to other existing machine learning, neural network, and transfer learning methods. The comparison demonstrates that our proposed approach outperforms other alternative options.

We organize rest of this study as follows: We begin with a review of some of the most recent state-of-the-art research studies on plant leaf disease detection problems in Section 2. Following that, Section 3 illustrates the approach we introduce, which uses a lightweight Deep CNN. Then, in Section 4, we show experiments implementing the proposed architecture. Afterwards, we discuss the evaluation of our experimental results in Section 5. Finally, in Section 6, we discuss and demonstrate possible future implications.

2 RELATED WORK

With the advancements of technology, usage of Artificial Neural Networks (ANNs) among researchers gain importance. Moreover, Convolutional Neural Networks (CNNs) are also used in various aspects of research. And after implementing these DL architectures, researchers begin to provide some successful aspects, including predictions of plant disease detection. Photographs of leaves in various states of health (sick and healthy) from distinct plants are used to assess the plant's likely state. Apart from that, transfer learning is a well-known and often used ML strategy where a model is created for a specific job, which is utilized as the basis of the model created for another task.

For an auto early identification of plant diseases, researchers [10] utilize SVM classification techniques and a hyperspectral imaging methodology. Babu et al. [7] proposes a feed-forward neural network approach with a backpropagation model for identifying different types of pests and illnesses in leaves. Backpropagation assists in the calculation of the stochastic gradient descent (SGD) parameters necessary to find an optimal function for ANNs.

Deep CNNs to solve this problem utilizing various datasets and a varying number of layers are reported by authors [13] for various plant leaf diseases. Lee et al. [5] discuss a comparable deep CNN method for various plant identification tasks using plant leaf images and variable amounts of data. In addition, deep CNNs can be used to identify plant diseases and pests. This method is used to identify pests and illnesses in tomato plants. The authors of [1] evaluate at 40 distinct research papers that used deep learning techniques and are applied to a variety of agricultural problems. Some techniques, such as the model reported in work [16], employ a 14-Layer CNN architecture to identify Multiple Sclerosis in human brains which includes Batch Normalization, Dropout, and Stochastic Pooling. It results an accuracy of 98.77%.

Recently, Zhang et al. [17] develop a 13-layer CNN using data augmentation techniques and stochastic gradient descent to learn certain high-level features for classifying fruit pictures with a result of 94.94% in the final trial. Also, the authors of [12] providing an overview of the most well-known traditional approaches for detecting plant diseases. These approaches include spectroscopic, imaging, and volatile profiling-based methods for detecting plant diseases.

In our proposed work, we train and assess Deep CNNs in order to develop a model for detecting plant diseases using a dataset of plant leaf imagery. The next section covers the fundamentals of the implemented models.

3 RESEARCH METHODOLOGY

In this section, we propose our idea for a CNN architecture to forecast plant leaf disease. We begin by creating a variant of the typical CNN framework by modifying some critical parameters and functions.

3.1 Convolutional Neural Network (CNN)

A CNN is a type of artificial neural network that is especially developed to analyze vector input and is used in image recognition and processing.

A neural network is a special architecture that mimics the activity of neurons in the brain. Traditional neural networks are not optimized for image processing, thus they have to be input images in low-resolution chunks. The "neurons" of a CNN are more like those of the frontal lobe, the part of the brain responsible for processing visual information in humans and other animals. The layers of neurons are organized in such a way that they span the whole visual field, eliminating the partial image processing issue that affects regular neural networks.

A CNN uses a mechanism similar to that of a multilayer perceptron, which is tuned for a faster processing rate. A CNN is made up of numerous convolutional layers, pooling layers, dropout layers, fully connected (FC/Dense) layers, and normalization layers, as well as an input layer, an output layer, and a hidden layer.

3.2 Proposed Methodology

Our primary objective is to have a minimal number of parameters by keeping the input shape as much lower it is possible with a

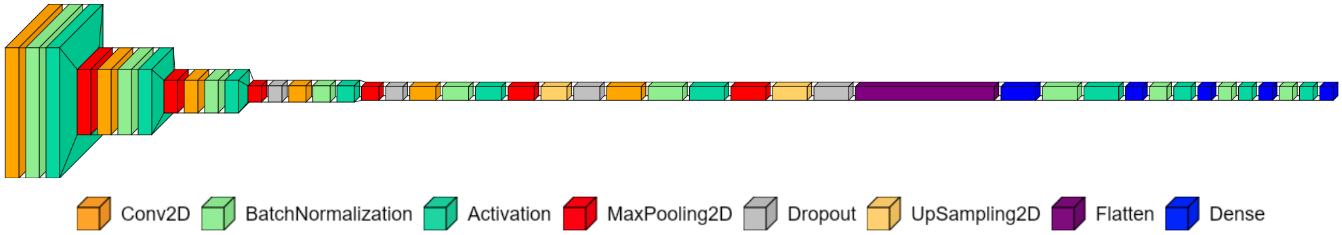

Figure 1: Visual Representation of Proposed Deep CNN Architecture

Figure 2: Picture of Leaves of Different Species and With Different States [From Left, Leaf Picture of Healthy Tomato, Potato Late Blight, Grape Leaf Blight, and Apple Cedar Rust]

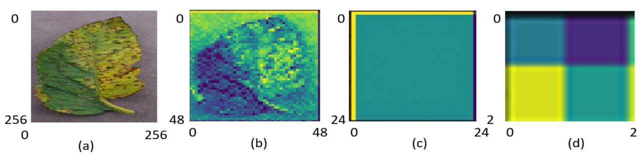

Figure 3: Feature Map

good result. A lower number of parameters in a model results in increased accuracy and computation speed. Our CNN model starts with a three-channel picture with a resolution of 48×48 pixels from the dataset. We use six Conv2D layers with (3,3) kernel size. Also, to avoid overfitting in the model, we use 10% dropout from the third Conv2D layer until the last one in our experiment. Moreover, we utilize MaxPooling layers with a pool size of (2,2) to minimize the computational cost of the layers. In the last two Conv2D layers, we use UpSampling layer to upscale our image and detect the anomaly better than usual and extract the feature. Additionally, we utilize default strides of (1,1) for all Conv2D layers.

We choose ReLU as the activation function because its gradient is not saturated, which considerably accelerates the evolving of stochastic gradient descent (SGD) compared to the other activation functions such as the Tanh / Sigmoid function. Finally, we convert the values to an one-dimensional array then we begin adding fully connected (FC) layers to the CNN, starting with 512 nodes, then 256 nodes, 128 nodes, and finally 64 nodes. In the output layer, we use the Softmax activation function as a network classifier. The softmax function is included towards the conclusion of the output since it is where all nodes can be categorized. The softmax classifier is a generalized binary variant of logistic regression. The model has been built utilizing the improved hyper-parameters.

We use Adam as the optimizer which helps to maximize the efficiency of production with 0.00001 learning rate. Also, we limit the model to run up to 50 epoch with batch size of 32.

After experimenting with various tweaks in the model, we come up with our Deep CNN architecture. This architecture provides the best performance while minimizing the number of parameters and therefore less computing resources.

Layers	Output Shape	Parameters
2D Convolutional Layer	(None, 48, 48, 64)	1792
Batch Normalization	(None, 48, 48, 64)	256
2D Max Pooling	(None, 24, 24, 64)	0
2D Convolutional Layer	(None, 24, 24, 96)	55392
Batch Normalization	(None, 24, 24, 96)	384
2D Max Pooling	(None, 12, 12, 96)	0
2D Convolutional Layer	(None, 12, 12, 128)	110720
Batch Normalization	(None, 12, 12, 128)	512
2D Max Pooling	(None, 6, 6, 128)	0
2D Convolutional Layer	(None, 6, 6, 256)	295168
Batch Normalization	(None, 6, 6, 256)	1024
2D Max Pooling	(None, 3, 3, 256)	0
2D Convolutional Layer	(None, 3, 3, 384)	885120
Batch Normalization	(None, 3, 3, 384)	1536
2D Max Pooling	(None, 1, 1, 384)	0
2D Up Sampling	(None, 2, 2, 384)	0
2D Convolutional Layer	(None, 2, 2, 512)	1769984
Batch Normalization	(None, 2, 2, 512)	2048
2D Max Pooling	(None, 1, 1, 512)	0
2D Up Sampling	(None, 2, 2, 512)	0
Flatten	(None, 2048)	0
Dense	(None, 512)	1049088
Batch Normalization	(None, 512)	2048
Dense	(None, 256)	131328
Batch Normalization	(None, 256)	1024
Dense	(None, 128)	32896
Batch Normalization	(None, 128)	512
Dense	(None, 64)	8256
Batch Normalization	(None, 64)	256
Dense	(None, 38)	2470
Total parameters		4,351,814
Trainable parameters		4,347,014
Non-trainable parameters		4,800

Table 1: Output Shape and Parameter Size of Each Layer of The Proposed Model

4 EXPERIMENTAL EVALUATION

Following the development of our suggested architecture, we test our model on various datasets and compare it against a number of existing deep learning architectures like VGG-16, Xception, and GoogLeNet, among others. Our primary objective is to create a model that is less in size and as much as more accurate possible than those distinct designs.

4.1 Dataset

Primarily, we use the PlantVillage Dataset[1] [4] which contains 38 distinct classes of plant leaf pictures. This collection of dataset includes 54,305 pictures. The images are of mainly sick and healthy leaves. These images are labeled with 38 distinct categories of leaves. And the dataset is divided into 80:10:10 training, validation, and test datasets, containing 43,449, 5,425, and 5,431 images, respectively.

4.2 Experimental Findings

We find many deep learning methods with an excellent; pretty competitive accuracy in worst-case scenario in Table 2. However, the parameter sizes of other approaches are enormous in comparison to ours. Then again, smaller models lack performance. In comparison to those models, our proposed model has a balanced parameter and high values for assessment metrics. The accuracy, precision, recall, F1, and loss measures are used in this study as the metrics.

Accuracy is defined as the ratio of the model's correct predictions to the total number of predictions made.

Precision is used to determine a proportion of correct identifications. Precision is calculated by dividing the number of true positive outcomes (TP) by the number of predicted positive outcomes (TP + FP). It is determined using the following equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1}$$

Recalls are used to figure out what percentage of true positives is appropriately detected. Recall is calculated by dividing the number of true positives (TP) by the total amount of data (TP + FN) is the important factor. The recall is calculated using the following equation:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2}$$

Similarly, F1 Score is a popular metric for assessing the effectiveness of machine learning algorithms. It is calculated as the arithmetic mean of accuracy and recall. The following equation is used to determine the F1 Score:

$$\text{F1} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

After using the dataset in our model, it achieves a really promising testing accuracy and other metrics in comparison to other available approaches and previous research. The one which outperforms our model is either more resource-hungry than our approach (in terms of parameters) or it is more of a hybrid approach that also take more computational power. The comparison is mentioned in Tables 2, 3, and 4.

From Table 4, we can see that most of the approaches use a good number of parameter whereas our model takes lesser than those approaches. Even if we use a bit higher number of epochs to train our model, it shows consistency and stability in the long run. And it is not problematic because we are saving the model for an end-to-end transfer learning approach that does not take any further training to run.

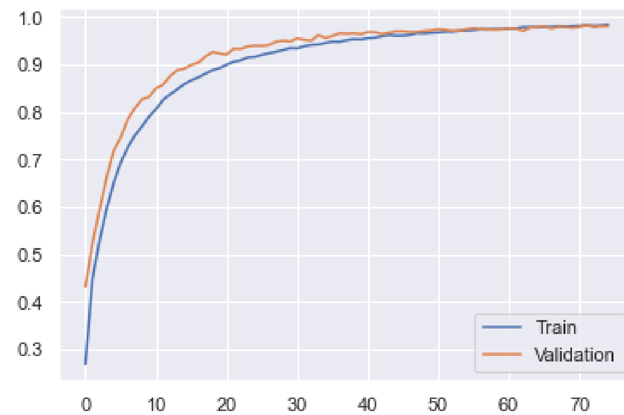

Figure 4: Accuracy Curve of The Proposed Model

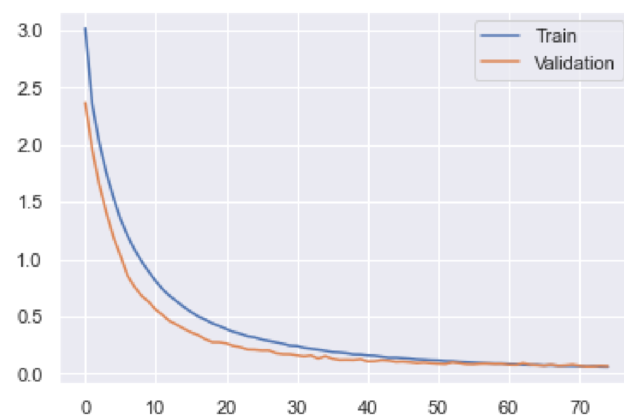

Figure 5: Loss Curve of The Proposed Model

5 DISCUSSION

We provide our findings on plant leaf disease using deep learning algorithms in the previous sections. According to our findings, models with fewer parameters yield lesser accuracy and performance. Moreover, our results indicate that employing many dense layers gives the model more excellent stability than before. Furthermore, when we try to create our model with fewer parameters, the model becomes increasingly inaccurate.

Talking about the models, The multiple dense layers in the fully connected layers, add more stability to our model. Also adding Batch normalization and dropouts make our model immune to huge overfitting issues. While working on the model, we see one of the unique characteristics that after putting serialized dense layers, it gives a boost in the performance of model.

And the evaluation metrics show our model is very balanced in terms of results and parameters, which is mentioned in Table 2 and 3 with the usage of lower resolution-based pictures. As a result,

[1] *https://github.com/spMohanty/PlantVillage-Dataset*

Deep Learning Architectures	Parameters (in Millions)	Epochs to Train the Model	Training Accuracy	Validation Accuracy
VGG-16	138	59	0.8339	0.8189
OverFeat	141.8	58	0.8995	0.8603
Inception ResNet v2	54.3	58	0.9551	0.9091
ResNet-50	23.6	55	0.9873	0.9423
MLCNN	78	57	0.9583	0.9402
Inception v4	41.2	59	0.9586	0.9489
Improved GoogLeNet	6.8	53	0.9829	0.9521
AlexNet	60	54	0.9689	0.9578
DenseNet-121	7.1	56	0.9826	0.958
MobileNet	3.2	47	0.9764	0.9632
Hybrid AlexNet with VGG (AgroAVNET)	238	54	0.9841	0.9649
ZFNet	58.5	47	0.9752	0.9717
Cascaded AlexNet and GoogLeNet	5.6	57	0.9931	0.9818
Xception	22.8	34	0.999	0.9798
Ours	4.35	75	0.9836	0.9806

Table 2: Accuracy Comparison after Training Different Deep Learning Architectures (PlantVillage) [11]

Deep Learning Architectures	Parameters (in Millions)	Validation Loss	Precision	Recall	F1-score
VGG-16	138	0.5651	0.8182	0.8194	0.8188
OverFeat	141.8	0.433	0.8592	0.8628	0.861
Inception ResNet v2	54.3	0.3047	0.9075	0.9105	0.9089
ResNet-50	23.6	0.1923	0.9351	0.9358	0.9354
MLCNN	78	0.182	0.9386	0.9411	0.9398
Inception v4	41.2	0.1828	0.941	0.9466	0.9438
Improved GoogLeNet	6.8	0.1038	0.9528	0.9539	0.9533
AlexNet	60	0.1298	0.9563	0.957	0.9566
DenseNet-121	7.1	0.1323	0.9581	0.9569	0.9575
MobileNet	3.2	0.109	0.9624	0.9612	0.9618
Hybrid AlexNet with VGG (AgroAVNET)	238	0.1078	0.9626	0.9674	0.965
ZFNet	58.5	0.1139	0.9746	0.9751	0.9748
Cascaded AlexNet and GoogLeNet	5.6	0.0592	0.9749	0.9751	0.975
Xception	22.8	0.0621	0.9764	0.9767	0.9765
Ours	4.35	0.0654	0.9831	0.9816	0.9816

Table 3: Testing Loss and Other Metrics Comparison after Training Different Deep Learning Architectures (PlantVillage) [11]

Author	Method	Testing Accuracy (%)
Mohanty et al. [8]	AlexNet & GoogleNet	99.27, 99.34
Geetharamani et al. [2]	Nine Layer CNN	96.46
Oyewola et al. [9]	DRNN	96.75
Sladojevic et al. [13]	Finetuned CNN	96.3
Li et al. [6]	Shallow CNN from VGG16 with SVM & RF	94
Ours	**PLeaD-Net (Proposed)**	**98.18**

Table 4: Testing Accuracy Comparison (PlantVillage) after Training Different Deep Learning Architectures

users can upload low-resolution-based images that do not require any costly mobile phone or excellent internet connection. And this image can be used to web-based deep learning solutions which can detect if the plant has disease or not.

6 CONCLUSION AND FUTURE WORK

Developing countries, which are dependant on agricultural economy, need to be very efficient in farming technologies and systems. This happens as technologies now have a major impact on the economy of such countries, specially the countries from the global south. Considering the financial strength of farmers in these countries, development of a lightweight technological method for plant disease identification can help the farmers through providing a quick identification process demanding a limited computational resource.

Therefore, in this study, we use leaf images for the purpose of predicting plant diseases using limited computational resource. To do so, we develop a lightweight Deep CNN architecture called PLeaD-Net. Our proposed Deep CNN architecture PLeaD-Net identifies 38 distinct classes of healthy and sick leaf images with a competitively good classification accuracy. The PLeaD-Net architecture contains fewer parameters than existing machine learning and deep learning model based alternatives to confirm that PLeaD-Net can work with limited resources. Nonetheless, PLeaD-Net also outperforms the other alternatives with its few parameters. To demonstrate the performance as well as to assess the consistency and reliability of PLeaD-Net, we test it over PlantVillage Dataset.

To make our study much more comprehensive, in the future, we will continue to collect new images of plants covering diversity over geographical regions, leaf development patterns, cultivation settings, image quality, and modes in order to expand the class and size of the dataset. Besides, our future research will concentrate on identifying plant illnesses in portions of the plant other than the leaves, such as fruits and flowers. Moreover, we look forward to test our approach in various datasets that contains various plant leaves from various continents around the globe. Furthermore, we are looking forward to dig into the uncertainty and to analyze the explainability of different CNN architectures on the relevant datasets, which is very significant as plant disease identification is a sensitive task and it needs to confirm as less risk factor as possible.

REFERENCES

[1] Alvaro Fuentes, Sook Yoon, Sang Kim, and Dong Park. 2017. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors (Basel)* 17, 9 (2017), 2022.

[2] Geetharamani and Arun Pandian. 2019. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* 76 (2019), 323–338.

[3] Song Han. 2017. *Efficient Methods and Hardware for Deep Learning*. Retrieved Sept. 30, 2021 from http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture15.pdf

[4] David. P. Hughes and Marcel Salathe. 2016. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv:1511.08060 [cs.CY]

[5] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. 2015. Deep-plant: Plant identification with convolutional neural networks. In *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE.

[6] Yang Li, Jing Nie, and Xuewei Chao. 2020. Do we really need deep CNN for plant diseases identification? *Comput. Electron. Agric.* 178, 105803 (2020), 105803.

[7] M. S. Prasad Babu & B.Srinivasa Rao. 2007. Leaves Recognition using back propagatiuon neural network-advice for pest & disease control on crops. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.110.3632&rep=rep1&type=pdf

[8] Sharada P Mohanty, David P Hughes, and Marcel Salathé. 2016. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* 7 (2016), 1419.

[9] David Opeoluwa Oyewola, Emmanuel Gbenga Dada, Sanjay Misra, and Robertas Damaševičius. 2021. Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. *PeerJ Comput. Sci.* 7, e352 (2021), e352.

[10] T Rumpf, A-K Mahlein, U Steiner, E-C Oerke, H-W Dehne, and L Plümer. 2010. Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance. *Comput. Electron. Agric.* 74, 1 (2010), 91–99.

[11] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. 2020. Plant disease classification: A comparative evaluation of convolutional neural networks and deep learning optimizers. *Plants* 9, 10 (2020), 1319.

[12] Sindhuja Sankaran, Ashish Mishra, Reza Ehsani, and Cristina Davis. 2010. A review of advanced techniques for detecting plant diseases. *Comput. Electron. Agric.* 72, 1 (2010), 1–13.

[13] Srdjan Sladojevic, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, and Darko Stefanovic. 2016. Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* 2016 (2016), 3289801.

[14] UN 2017. *Least Developed Countries (LDCs)*. Retrieved Sept. 30, 2021 from https://www.un.org/development/desa/dpad/least-developed-country-category.html

[15] G. Wang, Yu Sun, and Jianxin Wang. 2017. Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. *Computational Intelligence and Neuroscience* 2017 (2017).

[16] Shui-Hua Wang, Chaosheng Tang, Junding Sun, Jingyuan Yang, Chenxi Huang, Preetha Phillips, and Yu-Dong Zhang. 2018. Multiple sclerosis identification by 14-layer convolutional neural network with batch normalization, dropout, and stochastic pooling. *Front. Neurosci.* 12 (2018), 818.

[17] Yu-Dong Zhang, Zhengchao Dong, Xianqing Chen, Wenjuan Jia, Sidan Du, Khan Muhammad, and Shui-Hua Wang. 2019. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimed. Tools Appl.* 78, 3 (2019), 3613–3632.