Early Detection of Casava Plant Leaf Diseases using EfficientNet-B0

Vijayalata Y
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
vijaya@griet.ac.in

Nikhil Billakanti
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
nikhil222.billakanti@gmail.com

Karthik Veeravalli
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
karthikveervalli@gmail.com

Ashlin Deepa R. N
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
deepa.ashlin@gmail.com

Lokesh Kota
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
lokeshkota2101@gmail.com

***Abstract*— Image recognition plays a major role in everyday life applications like medical image analysis, gaming, surveillance and security, industrial automation, and more recently it has gained massive backing in the agricultural industry to identify plant diseases in crops. Plant diseases are a huge problem in agriculture and incorporating machine learning algorithms for their early detection will help better yields and save the farmers from loses. This paper entails the use of such machine learning algorithms to detect leaf diseases in the cassava plant. Cassava is one of the largest sources of carbohydrates for the continent of Africa. It is also very vulnerable to several plant diseases; this in turn threatens the food security of the continent. The present study is based on four of such diseases that affect the cassava yield namely, Cassava Bacterial Blight (CBB), Cassava Brown Streak Disease (CBSD), Cassava Mosaic Disease (CMD), and Cassava Green Mottle (CGM). In this research work, EfficientNet-B0 is proposed for the early detection of these diseases. The EfficientNet-B0 models outperform existing CNNs in terms of accuracy and efficiency while reducing parameter size and FLOPS by an order of magnitude. It is easier to detect disease at an early stage without the assistance of professionals, saving farmers both time and money. And our proposed model gave an accuracy of 92.6%.**

Keywords- plant diseases, image recognition, Deep- learning, EfficientNet-B0, agriculture.

I. Introduction

Cassava is the fourth most important food crop in developing countries, trailing only rice, maize, and wheat [1],[2]. Its leaves are quite high in protein content and they can also be stored over a long period, and hence serve as reserve food [1],[3].

Africa accounts for more than 60% of worldwide cassava production (182 of 298 million tons) [1]. It is the most widely cultivated crop in the country's southern regions, contributing significantly to the country's Gross Domestic Product (GDP) and providing a significant source of income for rural farmers. The majority of global cassava yield is also consumed by Africa is a staple for about 500 million people [4]. The domestic consumption of cassava in the continent is expected to rise, which in turn has created a greater demand for the crop.

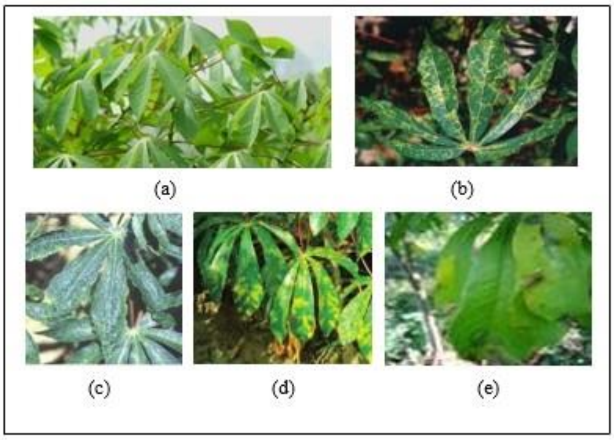

Fig. 1. (a) Healthy, (b) CMD, (c) CGM, (d)CBSD, (e) CBB

However, in 2020, cassava production had dropped by 6.7%. This was mainly due to an outbreak of Cassava Mosaic Disease(CMD). According to scientists, CMD is estimated to cause 15– 24% (equivalent to 12–23 million tons) of crop loss in Africa each year, which would equate to a loss of $1.2 to 2.3 billion USD [5]. Fig. 1, shows healthy and affected Casava leaves.

Cassava is drought tolerant, it can be grown on marginal land where other cereals fail, and requires minimal inputs, however, it happens to be very vulnerable to various viruses and other plant diseases [6]. CMD and Cassava brown streak disease (CBSD) is the most widespread viral disease, affecting at least half of Africa's cassava crops [7], [8]. These plant diseases cause significant reductions in the plant yield, resulting in significant economic losses.

II. Literature Study

In the age of climate change and globalization, accurate identification and diagnosis of plant diseases are critical for

2022 IEEE Delhi Section Conference (DELCON) | 978-1-6654-5883-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/DELCON54057.2022.9753210

food security and the control of the spread of exotic pests/pathogens.

Researchers have been striving to utilize machine learning techniques to aid farmers in the early detection of plant diseases. Some early works in the field have had promising results, but faced a few drawbacks of their own.

One of the leading works in recent times utilized a U-Net architecture for early detection of CMD and CBSD [9]. As UNet is very efficient while working with limited training samples and the results obtained to provide better performance for segmentation tasks. This architecture expands a vector to a segmented image using the same feature maps that are used for contraction (when converting an image into a vector) retaining the image's structural integrity while reducing distortion. However, when encountering deeper models, it faces the issue of slowing down in the middle layers [10]. Another study [11] proposed the use of MobileNetV2 [12] which makes the model highly effective in feature extraction and object detection. The results show that the model faces the issue of overfitting in the loss curves of training and validation data [13]. One other study in the field compared the performance of ResNet50, InceptionV3, and IncepetionResNetV2 architectures. Their accuracies were 78.29%, 83.01% and 84.77% respectively [14]. However, accuracy graphs showed large variations. Similarly, many such case studies that utilize the above architectures have shown that they have underperformed compared to EfficientNet-B0 models [15].

A Convolutional Neural Network (ConvNet/CNN) [16] is a Deep Learning system that is most commonly used for image analysis. CNN models have various filters/kernels which consist of trainable parameters that can convolve an image spatially, and identify features like edges and shapes. However, deep learning systems are hindered by a slow development rate. This could be improved with the use of transfer learning algorithms. Transfer learning is adaptable, allowing pre-trained models to be used directly for feature extraction, preprocessing, or integrating it into whole new models. This allows for faster progress or better performance while modeling a related task [17].

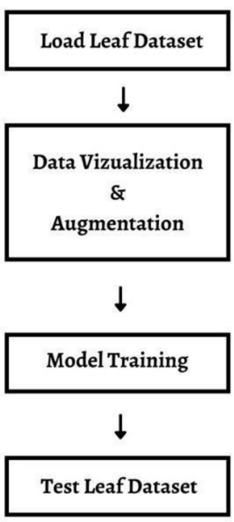

Fig. 2. The flow of the proposed methodology

EfficientNet-B0 [18] is a convolutional neural network architecture that belongs to the transfer learning framework. The EfficientNet-B0 model maximizes efficiency through the compound scaling method. EfficientNet-B0 was constructed using a multi-objective neural architecture search that maximizes accuracy as well as floating-point calculations [19]. A research paper on fresh fruit brunch ripeness classification has shown the implementation of deep learning with various transfer learning models such as EfficientNetB0, MobileNetV1 along with data augmentation. And the results have shown that EfficientNetB0 obtains an accuracy of 89.3% and MobileNetV1 with an 81.1%. The great difference in accuracy clearly shows the dominance of EfficientNet-B0 over other models [20].

In this [21] study conducted to identify and diagnose malaria parasite infections in blood smears the use of the EfficientNetB0 model has shown promising results with an accuracy of 94.70%. As EfficientNet-B0 is used in the above papers effectively, we propose the use of EfficientNet-B0 in our model.

III. METHODOLOGY

The approach used in this research work aims at early diagnosing the type of leaf disease the cassava plant has. The four types of diseases considered for this study are Cassava Bacterial Blight (CBB), Cassava Brown Streak Disease (CBSD), Cassava Mosaic Disease (CMD), and Cassava Green Mottle (CGM). The flow of the proposed methodology is given in Fig 2. The methodology is given in Algorithm 1.

The cassava leaf disease dataset was obtained from a Kaggle competition, which had a collective size of 5.76 GB out of which the training image dataset is of the size 2.38 GB [22]. It consists of 21,367 images in total, all classified into 5 classes, namely, Cassava Bacterial Blight (CBB) (1,087 images), Cassava Brown Streak Disease (CBSD) (2,189 images), Cassava Mosaic Disease (CMD) (13,158 images), Cassava Green Mottle (CGM) (2,386 images), and Healthy (2,577 images).

Algorithm 1:

Step 1: Load the Cassava leaf disease dataset from Kaggle.

Step 2: Split the dataset into training, validation, and test data.

Step 3: Analyse the data and apply data augmentation using Image Data Generator (If the dataset is imbalanced).

Step 4: Develop the EfficientNet-B0 model with average pooling before the output layer (Architecture is given in Fig.3).

Step 5: Pass the training data obtained in Step 3 to the model.

Step 6: Apply validation data and check the accuracy and loss for both training and validation data.

Step 7: Repeat step 6 to fine-tune the model parameters.

Step 8: Apply test data and calculate performance measures such as accuracy, precision, and recall.

Image augmentation uses a variety of processing techniques or a combination of techniques to create training images, including random rotation, shifts, shear, and flips, among others (techniques performed on already exiting images in the dataset). Due to a significant imbalance in the dataset (caused by dominance in the number of images for CMD disease), the proposed model makes use of the Image Data Generator tool [23] as part of the Image Augmentation process to artificially scale up the count for the other diseases, leading to balance in the dataset. The developed model makes use of EfficientNetB0 to facilitate transfer learning capabilities. The image from the balanced dataset passes into EfficientNet-B0 architecture (Fig. 3). EfficientNet-B0 uses a sequence of MBConv blocks in its architecture. Each image passes through various layers of this architecture for feature extraction and transforms according to the size of the filter.

Initially, the image is of size 224*224, it is given as input to a Convent layer on which a filter of the size 3*3 is applied, and it transforms the dimensions of the image to 224*224*32, which is easier to process in further layers without losing the features. The output of the convent acts as an input to MBConv1. Where a filter of size 3*3 is applied which now transforms the resolution from 224*224*32 to 112*112*16. Now the output of the feature map obtained from MBConv1 is passed on to 2 successive layers of MBConv6 where a filter of size 3*3 reduces the feature map of the image to 112*112*24.

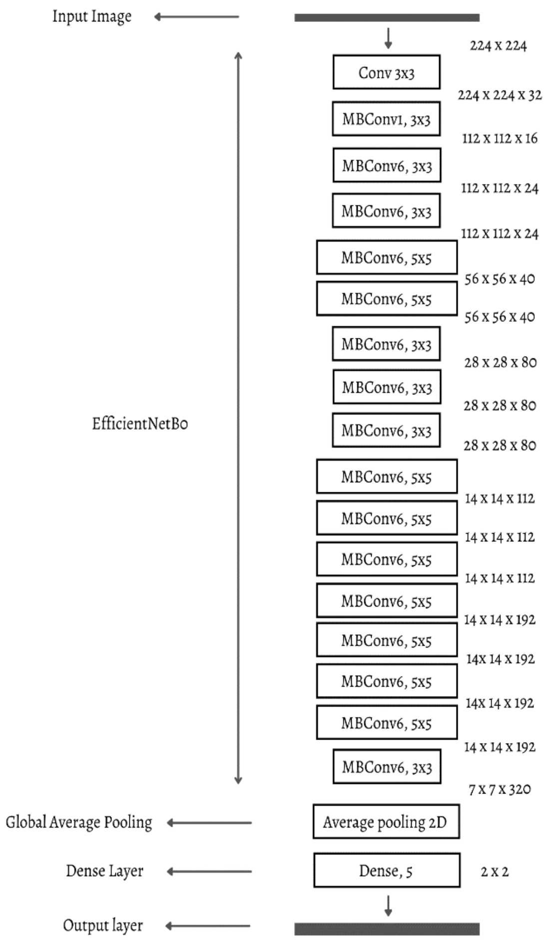

Fig. 3. EfficientNet-B0 Model Architecture

Following this, the output obtained is passed on to 2 layers of MBConv6 which have a filter size of 5*5, to reduce the feature map to 56*56*40. The result is then passed onto 3 consecutive layers of MBConv6 which have a filter of size 3*3 to transform the feature map size to 28*28*80. Now again the obtained image is fed onto 7 layers of MBConv6 which have a filter of size 5*5 to produce a feature map of size 14*14 and 192 channels. Finally, it is passed on to the last layer of MBConv6 which has a filter of size 3*3 and produces an output of size 7*7*320.

After passing the image through the EfficientNet-B0 layers, an average pooling [24] layer is introduced. When using the average pooling method, the color of the image is smoothed out. Max pooling [25] selects the image's brighter pixels. It's useful when the image's background is dark however this model is more interested in the lighter pixels. The choice of average pooling over max-pooling was because in max-pooling only the maximum of the stride (number of pixels in a matrix) is considered which leads to late recognition as many pixels are ignored. Whereas in average pooling, the average value of the stride is considered. A dense layer [26] is then used to classify images based on output from the convolutional layers. The activation function used in the output layer is softmax, rather than sigmoid. Sigmoid is used for binary classification whereas, softmax is a function that is used for multiclass classification and it converts a number vector to a probability vector [27]. The output values obtained from the softmax function resemble the probability of membership for each class (the 4 disease classes and 1 healthy leaf class). Once the model is defined, it must be compiled, and it is at this point that the loss function, optimizers, and metrics for prediction are defined. Sparse categorical cross-entropy is the loss function used and the evaluation criteria used in this model is accuracy. And the most important parameter that has to be defined in the compilation step is the optimizer, which adjusts the weights and learning rate to reduce overall loss and improve accuracy. Adam is the optimizer used in this model with a learning rate set to 0.001.

IV. Analysis

The original image dataset consists of 21,367 images. The dataset was split into 80% for training data, 20% for validation, and 20 samples are taken as test cases. An imbalance was observed in the dataset caused by the abundance of the images pertaining to the CMD class. This was tackled by using image augmentation techniques with aid of the Image Data Generator tool, which artificially increases the number of images in other classes. This augmented data was given for training.

In this current model, the total number of epochs is set to 10 with a batch size of 32. The main challenge with training neural networks is choosing the right number of training epochs as training data with a large number of epochs can lead to overfitting, while training data with a small number of epochs can lead to underfitting. To overcome this obstacle, the early stopping approach is employed. Early stopping is a technique for stopping training when the model's performance on a holdout validation dataset stops improving [28]. Validation loss is the monitored parameter with a patience level set to 5 epochs. ReduceLROnPlateau reduces the learning rate of a model when a learning metric has stopped improving. This often benefits the model as the learning rate is reduced by a factor of 2-10 when the learning stagnates for a patience number of epochs for the specified learning metric [29]. The learning rate of this model was set at 0.001. And the training and validation accuracy obtained for the proposed model is 89.04 % and 85.3 %, respectively. Fig.4 &5, demonstrate the amount of accuracy and losses during the training and validation procedure. The number of epochs is represented on the x-axis, while the accuracy and loss are represented on the y-axis. The model was run for 10 iterations. Fig.4 shows that the model obtains peak accuracy at epoch 8 and stagnates on further iterations. Also, Fig.5 shows that after epoch 8, the amount of loss stagnates. The results obtained after the model fed with the test cases come to be a total of 92.6% accuracy, which determines the percent of correct classification. The use of average pooling and data augmentation techniques helped the model attain higher levels of accuracy than the existing models.

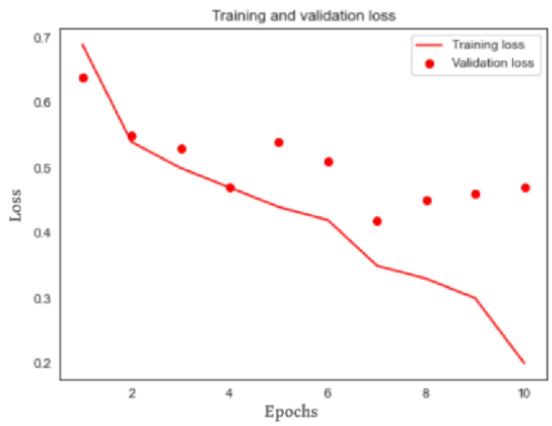

Fig. 4. Training and Validation accuracy

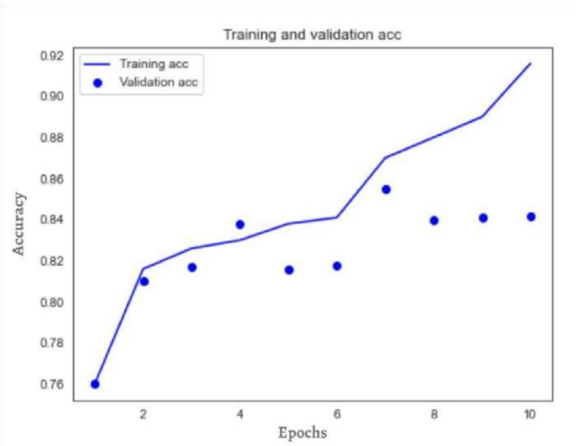

Fig. 5. Training and Validation loss

TABLE I. PRECISION, RECALL, AND F1-SCORE FOR PROPOSED MODEL

	Precision	Recall	F1-score
0	1.000	0.833	0.909
1	0.833	0.833	0.833
2	0.833	1.000	1.000
3	1.000	1.000	1.000
4	1.000	1.000	1.000

Table I. displays the precision, recall, and f1-score of the individual classes considered for classification. Precision and recall are performance measurements that apply to data retrieved from a collection. Table II shows the results obtained from research papers in the literature. It can be ascertained that EfficientNet-B0 produced a greater degree of accuracy compared to earlier investigations. EfficientNet-B0 uses fixed scaling coefficients for depth, width, and resolution. The number of layers is dependent on the size of the input image, the model requires more channels and a bigger receptive field if the input image is bigger.

TABLE II. COMPARISON OF PROPOSED METHOD WITH EXISTING CASSAVA LEAF DISEASE PREDICTION METHODS

Model	Accuracy (%)
MobileNetV2 [11]	74.5
U-Net [9]	83.9
ResNet50 [14]	78.29
InceptionV3 [14]	83.01
IncepetionResNetV2 [14]	84.77
VGG-16 [30]	85.5
EfficientNet-B0(Proposed model)	92.6

V. Conclusion

This study addresses the difficulty of identifying and classifying plant disease by analyzing cassava leaves. The main objective was to propose a method to classify cassava leaf diseases. The study proposed the use of the EfficientNet-B0 model [18] which is a state-of-the-art methodology. It is a scaling method that uses a compound coefficient to scale all depth/width/resolution dimensions uniformly. This method outperforms the existing research works on the early detection of cassava leaf disease classification. The current approach could help the farmers and the agricultural industry. From the experimental results, we found that the proposed model provides acceptable performance. The accuracy of the overall system reached 92.6%. In future works, this model could be improved to classify other cassava diseases, as well as other plant diseases.

References

[1] T. A.Ajayeoba, O. F. Ugbekile, and T. B. Hammed, "The use of cassava leaves as food and medicinal herbs in rural communities and the perceived health risks", Globalpresshub, pp. 19-38, June 1, 2021.

[2] Viduranga Y. Waisundara, "Cassava as a Staple Food", Researchgate.net, January 2018.

[3] Wivine Munyahali, Pieter Pypers, Rony Swennen, JeanWalangululu, BernardVanlauwe, RoelMerckx, "Responses of cassava growth and yield to leaf harvesting frequency and NPK fertilizer in South Kivu, Democratic Republic of Congo", ScienceDirect, Vol. 214, pp.194-201, December 2017.

[4] Dr.Leonardo, B. Carvalho, "Socio economic profile of cassava flour producers", CPSjournal, April 22,2020.

[5] “Protecting cassava, a neglected crop, from pests and diseases”, Food and Agriculture Organization of the United Nations, November 2019.

[6] CharlesOrek, WilhelmGruissem, MoragFerguson, HerveVanderschuren, "Morpho-physiological and molecular evaluation of drought tolerance in cassava", Science Direct, vol. 255,September 15,2020.

[7] Katie R. Tomlinson, Andy M. Bailey, Titus Alicai, Sue Seal, and Gary D. Foster, “Cassava brown streak disease: historical timeline, current knowledge and future prospects”, Molecular Plant Pathology, British Society for Plant Pathology, Vol. 19, Issue 5, pp. 1282-1294, December 2017.

[8] Joao Junior, "Cassava viruses threaten food security", Plantwise Blog, July 21, 2016.

[9] Patike Kiran Rao, R Sandeep Kumar and Dr. K Sreenivasulu, “Cassava Leaf Disease Classification using Separable Convolutions UNet”, Turkish Journal of Computer and Mathematics Education, Vol. 12, Issue 7, pp. 140-145, April 2021.

[10] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari, “Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation”, arxiv.org.

[11] H R Ayu, A Surtono and D K Apriyanto, “Deep learning for detection cassava leaf disease”, IOP Publishing Ltd, Phys.: Conf. Ser. 1751, September 2020.

[12] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks”, arxiv.org, 2019.

[13] H R Ayu, A Surtono and D K Apriyanto, “Deep learning for detection cassava leaf disease”, ResearchGate, Phys.: Conf. Ser. 1751, January 2021.

[14] Megha Mankal, Chinnapani Kiran Kumar and Dr. Samala Nagaraj, “Detection of Casava Plant related Diseases using Deep Learning”, International Research Journal of Plant Science, Vol. 12, Issue 3, pp. 01-06, June 2021.

[15] Mingxing Tan, Quoc Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, Proceedings of the 36th International Conference on Machine Learning, pp.6105-6114, 2019.

[16] Saad Albawi, Tareq Abed Mohammed, “Understanding of a Convolutional Neural Network”, The International Conference on Engineering and Technology 2017 At: Antalya, Turkey, August 2017.

[17] Jason Brownlee, “A Gentle Introduction to Transfer Learning for Deep Learning”, Machine Learning Mastery, December 2017.

[18] Mingxing Tan, Quoc Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”, arxiv.org, 2019.

[19] Deyuan Zhang, Zhenghong Liu, Xiangbin Shi, “Transfer Learning on EfficientNet for Remote Sensing image Classification”, IEEE, May 2021.

[20] Suharjito Suharjito, Gregorius Natanael Elwirehardja, Jonathan Sebastian Prayoga, “Oil palm fresh fruit bunch ripeness classification on mobile devices using deep learning approaches”, Computers and Electronics in agriculture, ResearchGate, September 2021.

[21] Francis Jesmar Perez Montalbo, Alvin Sarraga Alon, “Empirical Analysis of a Fine-Tuned Deep Convolutional Model in Classifying and Detecting Malaria Parasites from Blood Smears”, KSII Transactions on Internet and Information Systems, ResearchGate, January 2021.

[22] https://www.kaggle.com/c/cassava-leaf-disease-classification/data

[23] Connor Shorten and Taghi M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning”, Journal of Big Data, Article 60, July 2019.

[24] Vishal Passricha, Rajesh Kumar Aggarwal, in Intelligent Speech Signal Processing, “End-to-End Acoustic Modeling Using Convolutional Neural Networks”, ScienceDirect, 2019.

[25] Haibing Wu and Xiaodong Gu, "Max-Pooling Dropout for Regularization of Convolutional Neural Networks", arxiv.org.

[26] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q.Weinberger, “Densely Connected Convolutional Networks”, arxiv.org, January 2018.

[27] Binghui Chen, Weihong Deng, Junping Du, “Noisy Softmax: Improving the Generalization Ability of DCNN via Postponing the Early Softmax Saturation”, ResearchGate, July 2017.

[28] Rich Caruana, Steve Lawrence, Lee Giles, “Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping”, Proceedings of the 13th International Conference on Neural Information Processing Systems, pp.381-387, 2000.

[29] https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html

[30] Mohneesh S, “Cassava Leaf Disease Detection (2020) Writeup”, Analytics Vidhya, February 2021.