Rice Leaf Diseases Classification Using Discriminative Fine Tuning and CLR on EfficientNet

Susant Bhujel[1], Subarna Shakya[2]

[1]Tribhuvan University, Institute of Engineering, Pulchowk Campus, Department of Electronics and Computer Engineering, Pulchowk, Lalitpur
[2]Professor of Computer Engineering, Tribhuvan University, Institute of Engineering, Central Campus, Department of Electronics and Computer Engineering, Pulchowk, Lalitpur

E-mail: [1]076msice019.susant@pcampus.edu.np, [2]drss@ioe.edu.np

Abstract

Rice cultivation in Nepal is effect by many factors, one of the main factor is rice leaf diseases which limits the crops production. Image classification of rice leaf classify different rice leaf diseases. Image dataset of rice leaf diseases is taken from open source platform. Pre-processing of image is done which is followed by feature extraction and classification of images. This thesis presents image classification of rice leaf diseases into four classes, namely: Brown Spot, Healthy, Hispa, Leaf Blast using Convolutional Neural Network (CNN) architecture EfficicentNet-B0 and EfficicentNet-B3 based on fine-tuning with cyclical learning rate and based on discriminative fine-tuning. It is found that the test accuracy of EfficientNet-B0 is 81.96% and EfficientNet-B3 is 85.12% based on fine-tuning with cyclical learning rate and the test accuracy of EfficienNet-B0 is 83.99% and EfficientNet-B3 is 89.18% based on discriminative fine-tuning for 15 epochs. The results also conclude that the CNN architectures work better with discriminative fine-tuning than on fine-tuning with cyclical learning rate. The classification models of EfficicentNet-B0 and EfficicentNet-B3 are evaluated by recall, precision and F1-score metrics.

Keywords: Rice leaf diseases, convolutional neural network, EfficientNet, discriminative fine-tuning, fine-tuning, cyclical learning rate

1. Introduction

In Nepal, Rice is a significant crop that is expanding, making up around 50% of the nation's total agricultural land and production. Rice leaf infections are one of the factors that restrict the nation's rice production. These illnesses pose a serious threat to the quality and

DOI: https://doi.org/10.36548/jscp.2022.3.006
Received: 01.08.2022, received in revised form: 25.08.2022, accepted: 09.09.2022, published: 19.09.2022

yield of the crop, which will have an impact on global food supplies. Therefore, a disease's diagnosis comes first. Disease control interventions can be a waste of time and money and can result in additional losses without accurate identification of the disease and the disease-causing agent. One method for identifying infections is to classify images of rice leaves that have different types of diseases on them. CNN is frequently used class of ANN to study visual imagery when there is a topic of picture classification. The fundamental advantage of CNN over its forerunner is that it categorizes the photos without any human intervention and automatically marks the key aspects. LeNet-5 (1998), AlexNet (2012), VGG-16 (2014), Inception-v1 (2014), Inception-v3 (2015), ResNet-50 (2015), Xception (2016), Inception-v4 (2016), ResNeXt-50 (2017), Gpipe (2019), and EfficientNet (2019) are a few examples of CNN architectures. Over time, CNNs have improved significantly, largely as a result of increased processing power, experimentation, novel concepts, and the global deep learning community's intense interest. However, there is still a great deal to learn about these adaptable networks.

The role of the optimizer is introduced by the use of CNN architectures. Optimizers are methods or algorithms that are used to change the weights and learning rates of neural networks in order to reduce the loss function. Numerous optimizers exist, including adgrad, adaDelta, adam, stochastic gradient descent, mini-batch gradient descent, and gradient descent. They each have advantages and disadvantages of their own. Learning rate is a hyperparameter that may be customized and is used to train neural networks. It typically has a tiny positive value between 0.0 and 1.0. It is in charge of determining how quickly the model is adjusted to the issue. Small learning rates necessitate more training epochs, which increases the likelihood that the process may stuck, whereas large learning rates necessitate fewer training epochs, which increases the likelihood that the model will converge too soon to an unsatisfactory result. Accuracy can be increased in a few training epochs by carefully choosing the learning rate. It is the model's most crucial hyperparameter. On CNN, a variety of learning rate scheduling methods are available, including stochastic gradient descent with warm restarts, differential learning rate, cyclical learning rate, step decay learning rate, exponential decay learning rate, and constant learning rate.

2. Related Work

The classification of rice leaf diseases has been the subject of extensive research, which demonstrates the application of methods and theories to the findings of the study. Rice leaf diseases classification with three class (brown spot rice disease, leaf smut rice disease, bacterial leaf blight disease) was carried out based on image processing technology such as random forest, decision tree, gradient boosting and naïve-baye classification algorithms [1]. The best result of performance was given by random forest algorithm equal to 69.44%. Image classification of rice leaf diseases with four class (leaf blast, leaf blight, brown spot, healthy) using VGG-16 CNN architecture was done in paper [2]. The accuracy of the model was 92.46%. Image-based plant disease detection [3] was done using deep learning to classify plant leaf into four class (bacterial spot, yellow leaf curl virus, late blight and healthy leaf). The features extracted were fitted into the neural network with 20 epochs.

EfficientNet [4] is a CNN architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. EfficientNet achieve much better and efficiency than its predecessor CNN architectures. Paper [5] shows the use of efficientNet architecture for remote sensing image classification based on transfer learning which outperforms than recent deep learning methods and significantly reduces training times and computing power as well.

Learning rate is the most important hyperparameter to tune for training deep neural networks. Cyclical learning rates practically eliminates the need to experimentally find the best values and schedule for the global learning rates. Instead of monotonically decreasing the learning rate, this method lets the learning rate cyclically vary between reasonable boundary values [6]. On comparative analysis [7] between various learning rate scheduling techniques on CNN, cyclical learning rate mechanism has performed better and produced accuracy at a much lesser epoch than the other techniques. So it is time and space-efficient than others.

Discriminative fine-tuning is a fine-tuning process in which all layers of model are tune with different learning rates instead of using same learning rate for all layers. This strategy is used in paper [8] for natural language processing. It helps in training better model for classification.

3. Proposed Work

3.1 System Block Diagram

The proposed method block diagrams for the rice leaf diseases classification is shown in the figures below:

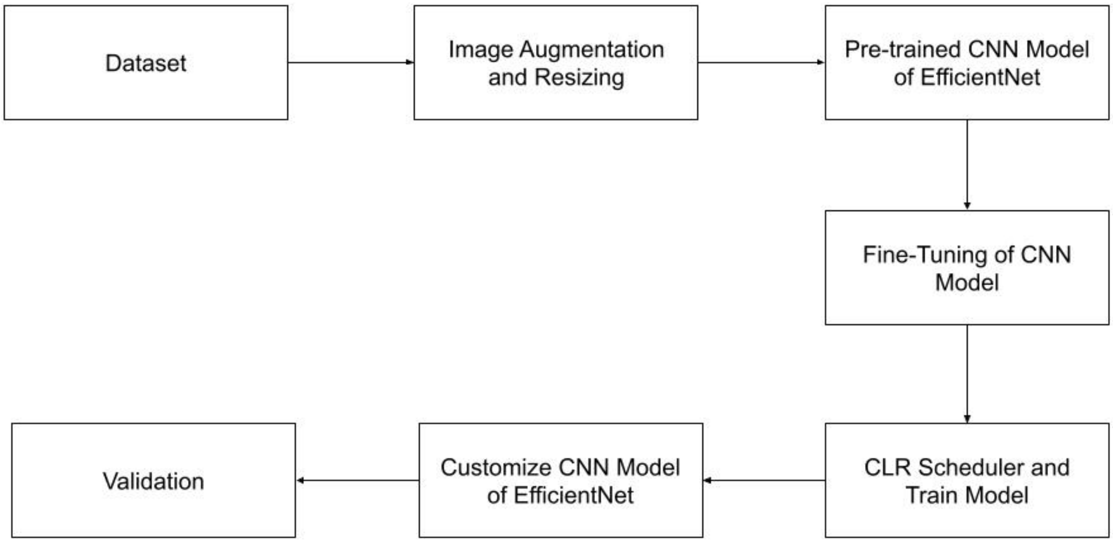

Figure 1. Proposed CLR Scheduler Based Rice Leaf Diseases Classification Model

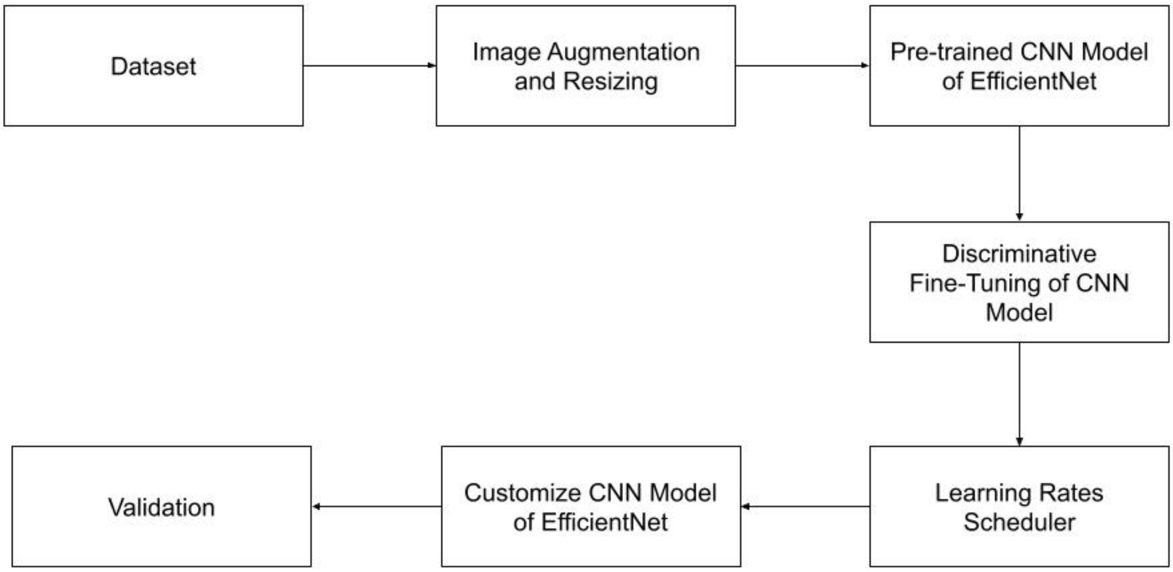

Figure 2. Proposed Discriminative Fine-Tuning Based Rice Leaf Diseases Classification Model

3.2 Dataset Description

Dataset is an essential part of this thesis. For this thesis, datasets is taken from open source site https://www.kaggle.com/datasets/jonathanrjpereira/rice-disease. This dataset contains images of four different classes of rice leaves. They are Healthy, Leaf Blast, Hispa and Brown Spot each with 523 images.

Figure 3. Healthy Data Sample

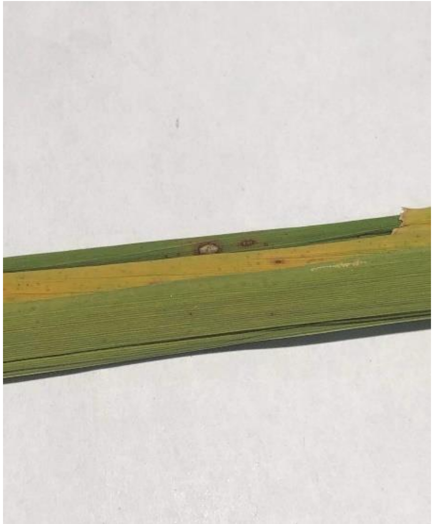

Figure 4. Leaf Blast Data Sample

Figure 5. Hispa Data Sample

Figure 6. Brown Spot Data Sample

3.3 Image Augmentation

By adding slightly changed versions of already existing data or by creating new synthetic data from existing data, image augmentation is a technique used to expand the amount of data. For picture augmentation, methods include twisting, shearing, earlier processing, zooming, rotation, etc. When a model is being trained, it works as a regularizer and helps minimize overfitting.

Image augmentation techniques like transpose, rotation, transpose with rotation are used to increase amount of data. So, the total data of images is 4*4*523=8368.

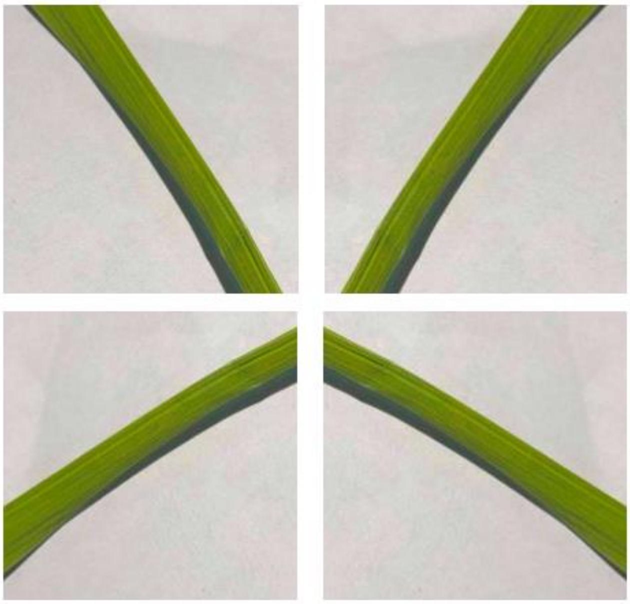

Figure 7. Image Augmentation of Healthy Data Sample

3.4 Image Preprocessing

Image dataset from https://www.kaggle.com/datasets/jonathanrjpereira/rice-disease contains images size of different width and height. By image preprocessing, images on dataset are made to the sizes of 224*224 and 300*300.

3.5 EfficientNet Family

EfficientNet is a family of CNN architecture from EfficientNet-B0 to EfficientNet-B7. EfficientNet-B0 is the baseline network of EfficientNet family. EfficientNet-B0 main building block is mobile inverted bottleneck (MBConv). In this baseline network using compound scaling method to scale depth, width and resolution to get EfficientNet-B1 to EfficientNet-B7. EfficientNet uses a technique called compound coefficient to scale up models in a simple but effective manner.

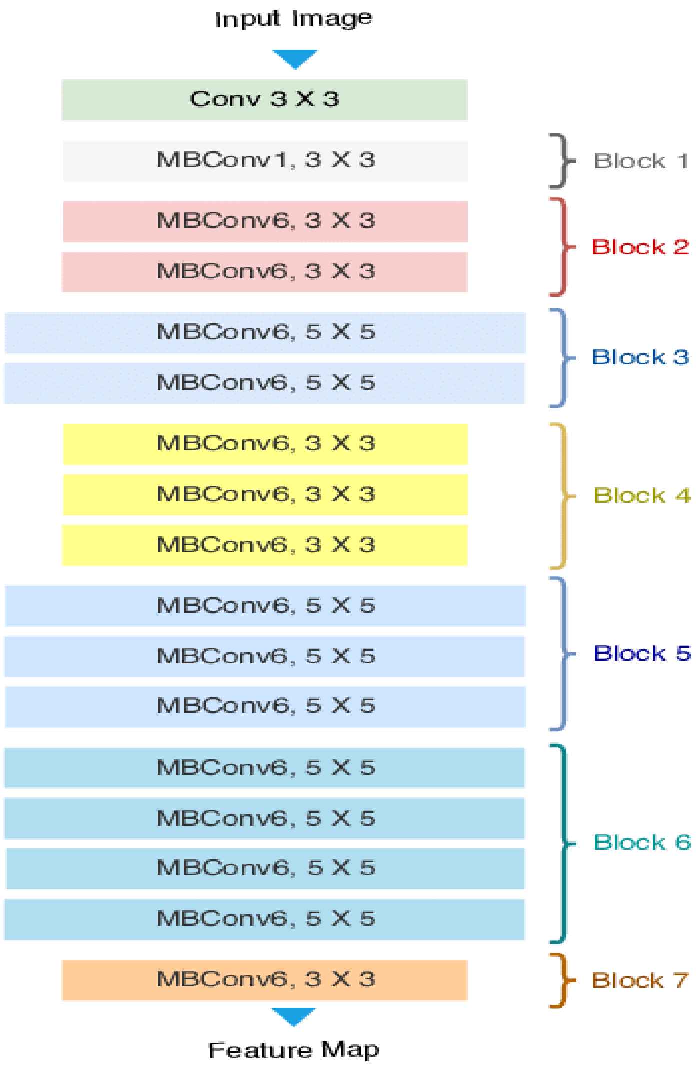

Figure 8. Architecture of EfficientNet-B0 with MBConv as Basic building blocks

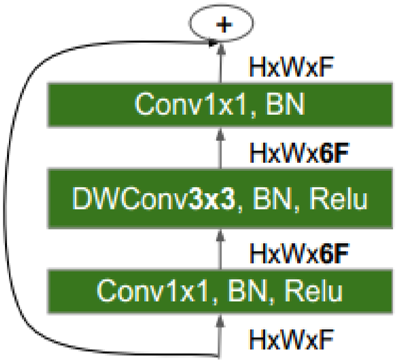

Figure 9. MBConv6, 3x3 [9]

3.5.1 EfficientNet-B0

EfficientNet-B0 is one the architecture of EfficientNet family. The size of image that this architecture input layer require is 224*224. This architecture have 237 layers.

3.5.2 EfficientNet-B3

EfficientNet-B3 is the fourth architecture of EfficientNet family. The size of image that this architecture input layer require is 300*300. This architecture have 384 layers.

3.6 Fine Tuning

The top 40 layers of EfficientNet-B0 and the top 30 layers of EfficientNet-B3 are retrained while the batchnorm of layers, which is trained on a large number of samples of data from Imagenet, is frozen. This refines the pre-trained EfficientNet architecture of B0 and B3. The key advantages of fine-tuning are increased productivity and reduced resource usage while developing new models. Because the machine learning model has already been trained, fine-tuning enables it to be built with relatively minimal training data.

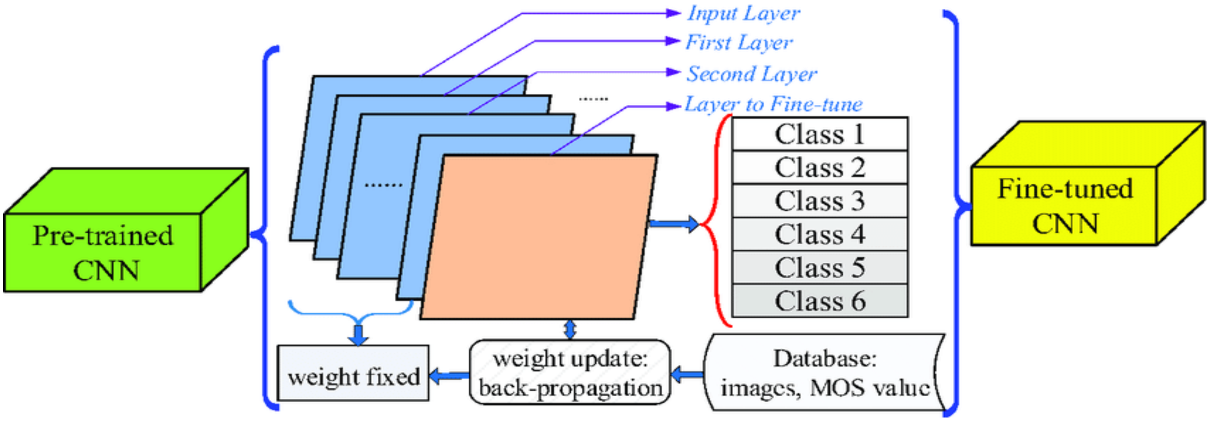

Figure 10. Fine-Tuning of CNN Architecture

3.7 Cyclical Learning Rate

Gradient descent may become caught in a local minimum during model training as opposed to a global minimum. However, in the case of cyclical learning rate, learning rate will grow in a cycle, taking out of local minimum if got stuck to local minimum. The other half of a cycle, which explores the more confined regions of the loss function, will see a decline in learning rate. As a result, the cyclical learning rate converges more quickly, producing accuracy at a substantially smaller number of epochs.

In this research, Cyclical Learning Rate having maximum learning rate (max_lr) = 1e-2, minimum learning rate (lr) = 1e-6, scale function = lambda x: 1 / (2.**(x-1)) , step size = 2*steps pre epoch is used with EfficientNet-B0 and EfficientNet-B3 CNN architecture. Here, steps pre epoch is equal to number of training data divided by batch size. This process is done using Google Colab, Tensorflow and Keras.

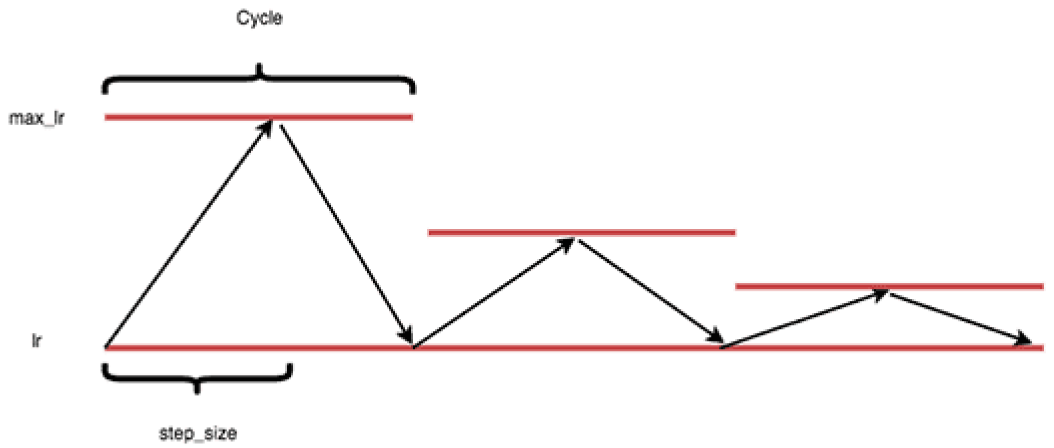

Figure 11. Triangular2 Schedule Cyclical Learning Rate

3.8 Discriminative Fine Tuning

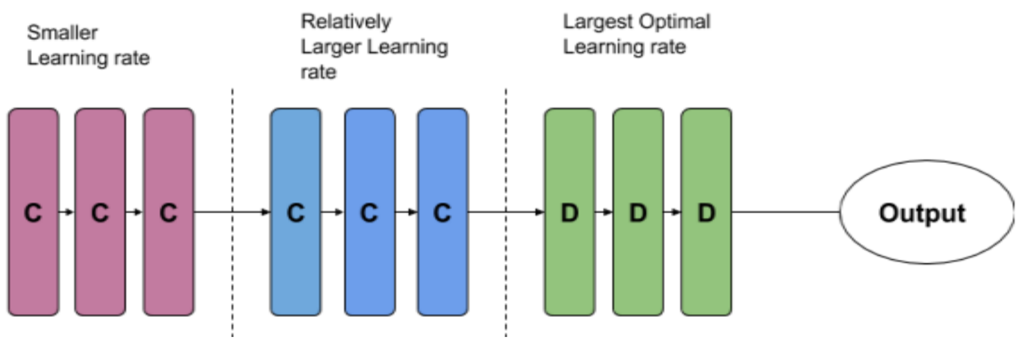

Figure 12. Discriminative Fine Tuning

Unfreezing all the layers of pre-trained model of EfficientNet-B0 and EfficientNet-B3 CNN architectures are fitted with learning rate in the interval of 6.31e-7 and 1e-2. Initial layer uses 6.31e-7 as learning rate and as the layer goes deeper learning rate goes in increasing to 1e-2. This process is done using Google Colab, Fastai and Pytorch.

4. Results and Discussion

Model for classification of rice leaf diseases are built by EfficientNet-B0 and EfficientNet-B3 CNN architecture based on fine-tuning with cyclical learning rate scheduler and based on discriminative fine-tuning using 6024 images for training with batch size of 64 for 15 epochs.

For validation and testing of trained models 670 and 1674 images are used respectively. The test accuracy of EfficientNet-B0 is 81.96% and EfficientNet-B3 is 85.12% based on fine-tuning with cyclical learning rate scheduler and the test accuracy of EfficientNet-B0 is 83.99% and EfficientNet-B3 is 89.18% based on discriminative fine-tuning.

4.1 EfficientNet-B0 results based on fine-tuning with CLR Scheduler

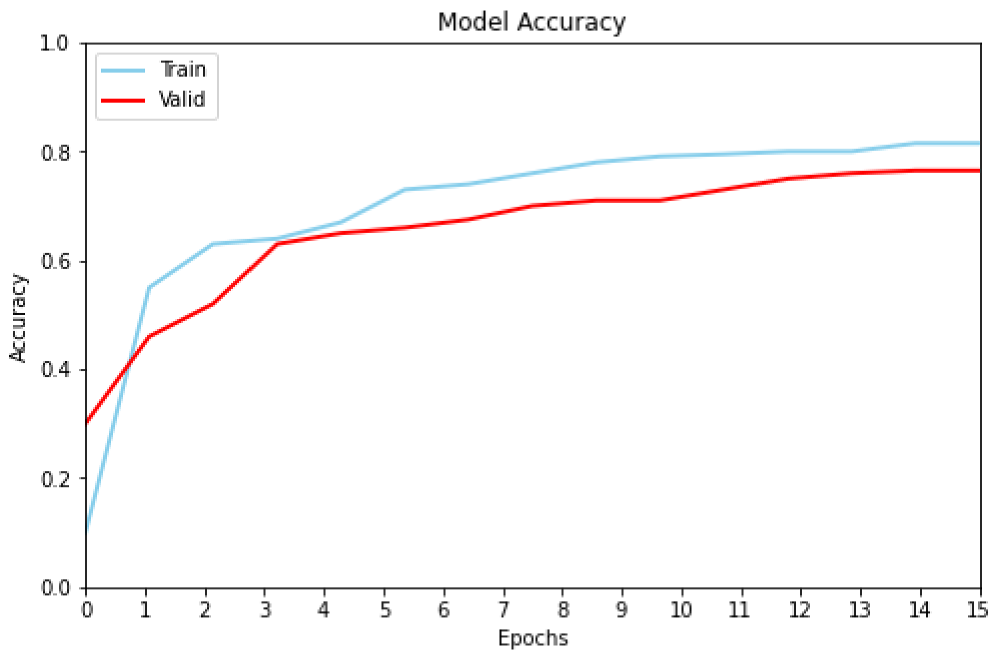

Figure 13. EfficientNet-B0 Model Accuracy

From Figure 13, we can see that the Train Accuracy and Valid Accuracy is 81.33% and 76.49% respectively at 15th epoch.

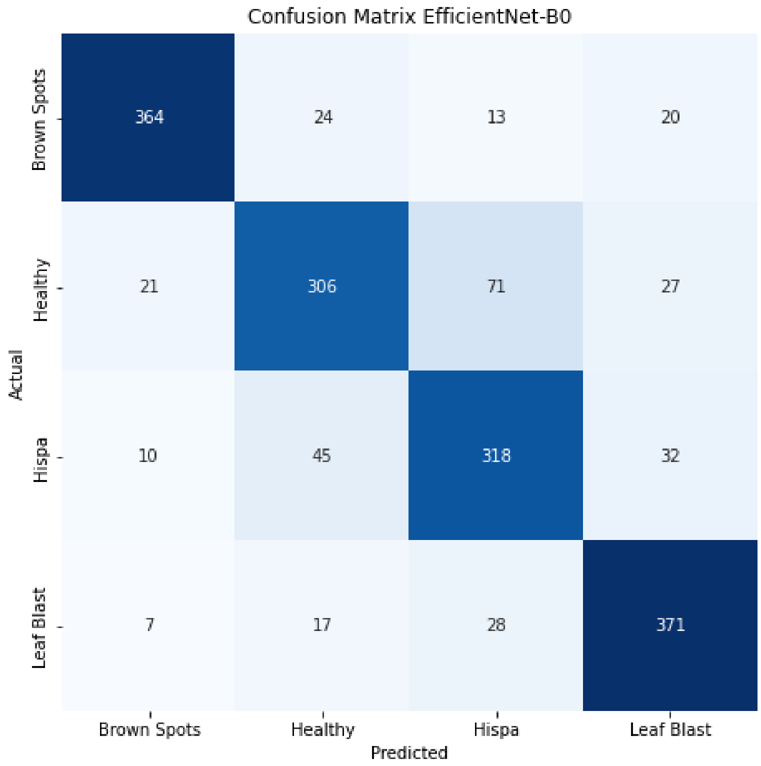

Figure 14. EfficientNet-B0 Confusion Matrix

Table 1. Performance Metrics of EfficientNet-B0 Model

Class	**Recall**	**Precision**	**F1 score**
Brown Spot	0.86	0.91	0.88
Healthy	0.72	0.78	0.75
Hispa	0.79	0.74	0.76
Leaf Blast	0.88	0.82	0.85

4.2 EfficientNet-B3 results based on fine-tuning with CLR Scheduler

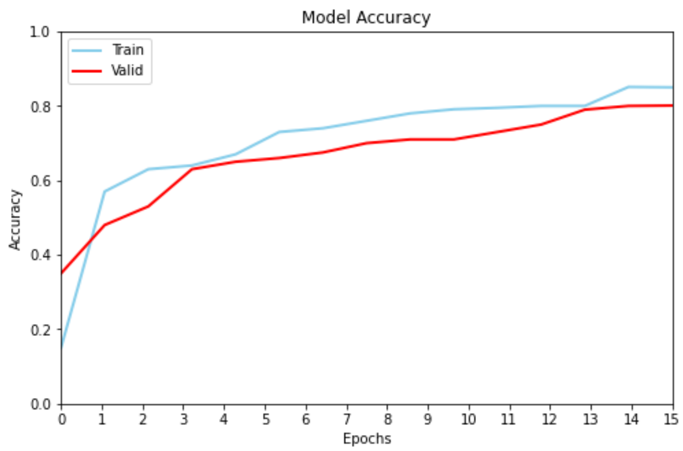

Figure 15. EfficientNet-B3 Model Accuracy

From Figure 15, we can see that the train accuracy and valid accuracy is 85.91% and 80.04% respectively at 15th epoch.

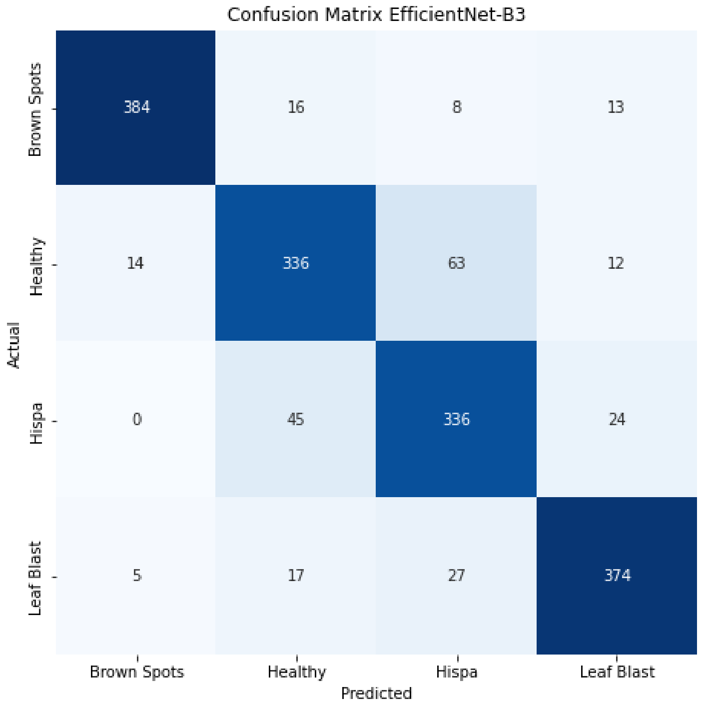

Figure 16. EfficientNet-B3 Confusion Matrix

Table 2. Performance Metrics of EfficientNet-B3 Model

Class	**Recall**	**Precision**	**F1 score**
Brown Spot	0.91	0.95	0.93
Healthy	0.79	0.81	0.80
Hispa	0.83	0.77	0.80
Leaf Blast	0.88	0.88	0.88

4.3 EfficientNet-B0 results based on Discriminative Fine Tuning

Table 3. Performance Metrics of EfficientNet-B0 Model

Class	Recall	Precision	F1 score
Brown Spot	0.91	0.97	0.93
Healthy	0.74	0.79	0.76
Hispa	0.87	0.69	0.76
Leaf Blast	0.80	0.92	0.85

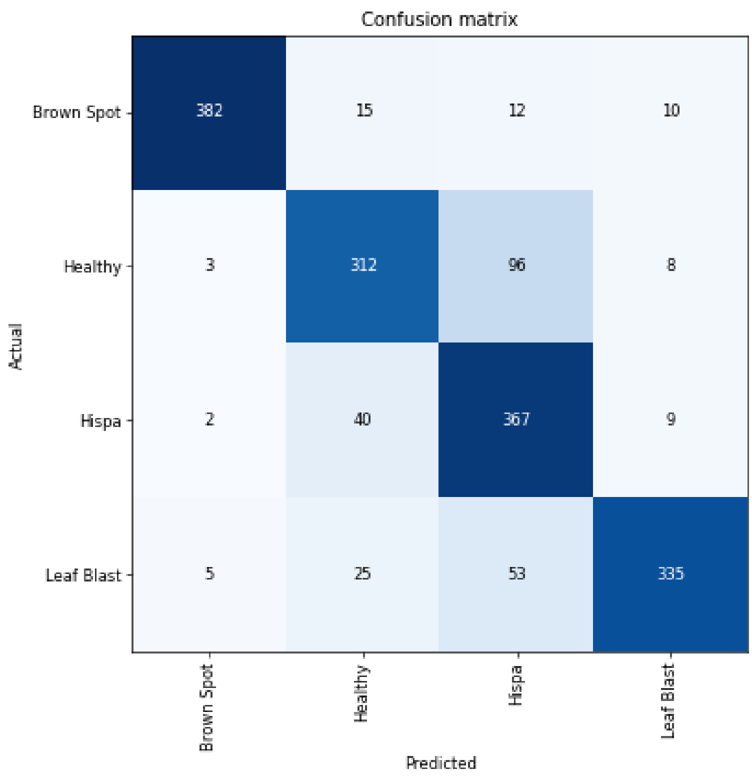

Figure 17. EfficientNet-B0 Confusion Matrix

4.4 EfficientNet-B3 results based on Discriminative Fine Tuning

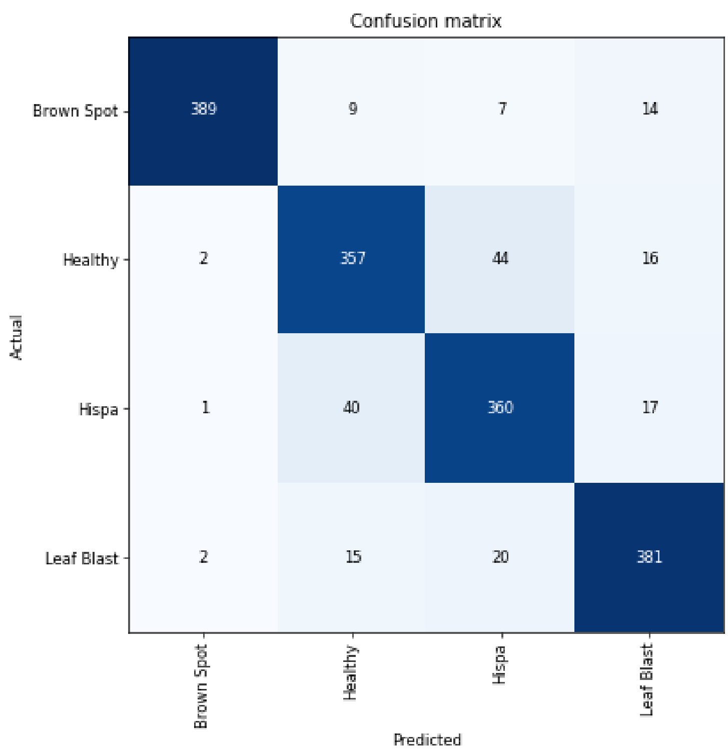

Figure 18. EfficientNet-B3 Confusion Matrix

Table 4. Performance Metrics of EfficientNet-B3 Model

Class	Recall	Precision	F1 score
Brown Spot	0.92	0.98	0.94
Healthy	0.85	0.84	0.84
Hispa	0.86	0.83	0.84
Leaf Blast	0.91	0.89	0.89

4.5 Software Requirement

4.5.1 Google Colab

A Google cloud-based service named Google Colaboratory that replicates Jupyter Notebook in the cloud is also known as Colaboratory. Nothing needs to be installed on our computer to utilize it. In many ways, using Colaboratory is similar to using a Jupyter Notebook desktop installation. The main audience for Google Colaboratory is readers who don't utilize a typical desktop configuration to work through the examples.

4.5.2 Tensorflow

One of the complete open source machine learning platform is TensorFlow. Researchers can advance the state-of-the-art in machine learning thanks to its extensive, adaptable ecosystem of tools, libraries, and community resources, while developers can simply create and deploy ML-powered applications.

4.5.3 Keras

The Keras API is created with people in mind, not with computers. Best practices for lowering cognitive load are followed by Keras, which provides consistent & simple APIs, reduces the amount of user activities necessary for typical use cases, and offers clear & actionable error signals. Additionally, it contains a wealth of developer instructions and documentation.

4.5.4 Pytorch

Developed primarily by Meta AI, PyTorch is an open source machine learning framework that is based on the Torch library and is used for tasks like computer vision and natural language processing. It is open-source software that is available for free under the

Modified BSD license. PyTorch features a C++ interface, even though the Python interface is more refined and the main focus of development.

4.5.5 Fastai

FastAI is a deep learning library that gives practitioners high-level components that can efficiently and quickly offer cutting-edge results in common deep learning domains and gives researchers low-level components that may be mixed and combined to create new methods. With minimal sacrifices, it seeks to achieve both usability and flexibility or performance.

5. Conclusion

In this research, Models are built with CNN architecture namely EfficientNet-B0 and EfficientNet-B3 for rice leaf diseases classification based on Fine-Tuning of pre-trained model and Cyclical Learning Rate (CLR) Triangular2 scheduler and based on Discriminative Fine-Tuning. Accuracy of models are obtained. Accuracy are 81.96% and 85.12% with EfficientNet-B0 and EfficientNet-B3 respectively based on CLR Scheduler and Accuracy with Discriminative fine tuning of EfficientNet-B0 and EfficientNet-B3 is 83.99% and 89.18% respectively. From the results, we can conclude to discriminative fine tuning gives the best result than on Fine-Tuning with CLR scheduler.

Future extension to this research includes, training the CNN architecture with large dataset of Rice leaf diseases, tuning of hyperparameter such as learning rate, batch size, etc , implementation of others CNN architectures such as EfficietnNetV2.

References

[1] Mekha, P., Teeyasuksaet, N. (2021) *Image Classification of Rice Leaf Diseases Using Random Forest Algorithm*, in 2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, IEEE

[2] Ghosal, S., Sarkar, K. (2020) *Rice Leaf Diseases Classification Using CNN With Transfer Learning*, in 2020 IEEE Calcutta Conference (CALCON), IEEE..

[3] Dhakal, A., Shakya, S. (2018) *Image-Based Plant Disease Detection with Deep Learning*, in International Journal of Computer Trends and Technology (IJCTT) – Volume 61 Number 1.

[4] Tan, M., Le, Q.V. (2020) *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, arXiv:1905.11946v5.

[5] Zhang, D., Liu, Z., Shi, X. (2020) *Transfer Learning on EfficientNet for Remote Sensing image Classification*, in 2020 IEEE 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), IEEE.

[6] Smith, L.N. (2017) *Cyclical Learning Rates for Training Neural Networks*, arXiv: 1506.01186v6.

[7] Konar, J., Khandelwal, P., Tripathi, R. (2020) *COMPARISON OF VARIOUS LEARNING RATE SCHEDULING TECHNIQUES ON CONVOLUTIONAL NEURAL NETWORK*, in 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science, IEEE.

[8] Howard, J., Ruder, S. (2018) Universal Language Model Fine-tuning for Text Classification, arXiv:1801.06146v5.

[9] Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V. (2019) *MnasNet: Platform-Aware Neural Architecture Search for Mobile*, arXiv: 1807.11626v3.

Author's biography

Susant Bhujel currently pursuing his MSc. Degree in Information and Communication Engineering from Pulchowk Campus, Institute of Engineering, Tribhuvan University.

Subarna Shakya received MSc and PhD degrees in Computer Engineering from the Lviv Polytechnic National University, Ukraine, in 1996 and 2000, respectively. Currently, he is a professor at the Department of Electronics and Computer Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University. His research interest includes e-government system, distributed and cloud computing, and software engineering and information system, Deep Learning, and Data Science.