

## Enhanced Classification of Sugarcane Diseases Through a Modified Learning Rate Policy in Deep Learning



Swapnil Dadabhau Daphal<sup>1\*</sup>, Sanjay M. Koli<sup>2</sup>

<sup>1</sup> Department of E&TC Engineering, G H Raisoni College of Engineering and Management Wagholi, Pune 412207, Maharashtra State, India

<sup>2</sup> Department of E&TC Engineering, Ajeenkya DY Patil School of Engineering, Charholi Bk., Pune 412105, Maharashtra, India

Corresponding Author Email: [swapnil.daphal.phdetc@ghrcem.raisoni.net](mailto:swapnil.daphal.phdetc@ghrcem.raisoni.net)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.410138>

### ABSTRACT

**Received:** 26 May 2023

**Revised:** 9 October 2023

**Accepted:** 10 November 2023

**Available online:** 29 February 2024

#### Keywords:

*deep learning, learning rate, plant diseases, sugarcane, small databases, mobilenet-v2, learning rate decay*

Motivated by the agriculture-centric economy of India, and specifically the challenges experienced in the sugarcane sector due to reduced yields from diseases including rust, red rot, yellow leaf, and mosaic, this study aims to harness effective deep-learning technologies for improved plant disease monitoring. The challenge of mitigating over-fitting, particularly when dealing with small datasets, is addressed through hyper-parameter tuning. In this study, we introduce an innovative modification to the learning rate decay policy, tested on a uniquely constructed small-sized database of sugarcane leaf images. This database encompasses five classes: healthy, rust, red rot, yellow leaf, and mosaic. To evaluate the effectiveness of the proposed learning rate policy, it was compared against multiple benchmark datasets and found to surpass established methods in performance metrics. This study introduces an additional exponential component into the learning rate policy to facilitate model convergence within the same number of epochs, thereby enhancing its performance over step, exponential, cosine, and exponential sine methods. A marginal improvement in scores was observed with the integration of the proposed learning rate policy and MobileNet-V2 as the backbone architecture. Remarkably, the MNIST dataset achieved a score of 99.9%, CIFAR-10 scored 92%, and the newly introduced database secured a score of 89%. These results underscore the efficacy of the proposed approach in enhancing the classification of sugarcane diseases.

## 1. INTRODUCTION

The cost function serves as an indicator of the model's learning efficiency and contributes to identifying the most suitable parameters for the training model. The tuning of these parameters is integral to the learning process, aiming to optimize the model's results. The model's learning capability improves as the loss function minimizes, propelling the model towards optimization [1, 2]. The cost function selected is contingent upon the training data and the specific problem being addressed. The primary objective of gradient-based approaches is to minimize the gradient of the cost function during the training phase [3-5]. However, this method encounters several obstacles when dealing with larger datasets, more complex models, and the increased prevalence of saddle points in the cost function space. Simple gradient descent-based algorithms struggle to converge in areas where the gradient approaches zero, a phenomenon known as the vanishing gradient problem. To counteract this issue, gradient-based methods that allow learning even at zero gradient points have been developed [6]. The momentum-based method addresses the vanishing gradient problem by adding the rate of change of the cost function to the relevant variable. In essence,

the model's learning pace is rapid when the rate of change in the cost function is high, and conversely, the model undergoes a slower training process when the rate is low [7]. Continuous learning is ensured by the previously summed value of the cost function, even if it reaches zero. However, it remains challenging to predict the optimal position of the cost function when learning concludes or parameters cease to change [8]. These issues are exacerbated when the database used for experimentation contains exceptionally small sample sizes.

The learning rate or step size plays a pivotal role in the optimization process of the gradient-based technique [9-11]. If the gradient is too small, learning proceeds at a sluggish pace, and if the gradient is excessively large, the model may not converge at all. To manage these issues, learning rate policies have been instituted. A constant learning rate policy keeps the learning rate (LR) steady, adjusting it based on the gradient. To overcome this limitation, strategies have been developed that allow the LR to change as the model learns (step-based and time-based). Moreover, adaptive LR methods have been established for improved learning outcomes [12-14]. Cyclic learning rate policies have leveraged the benefits of the warm restart, which periodically adjusts the LR to enhance performance [15-17]. Warm restart refers to the occasional

increment of the learning rate during training, typically used to evade local minima within the context of a cyclic learning rate policy. This strategy enables the model to explore a broader range of options before once again decelerating the learning rate.

While these approaches demonstrate superior performance with large and well-structured datasets, many practical applications rely on specialized databases with fewer samples. Evaluating the performance of machine learning use cases on such real-time data always presents an intriguing challenge. Frequently changing model architecture requires careful consideration of available resources. It is generally preferable to achieve superior results by optimally adjusting hyperparameters without imposing additional resource burdens. In this study, we introduce a modified learning rate policy that effectively manages model performance with smaller databases. The experiment is conducted on a sugarcane leaf disease database created by the authors. Baseline experiments are also carried out with MNIST and CIFAR-10 datasets to verify the efficacy of the adjusted learning rate policy. An in-depth exploration of the comparative study of the learning rate policy experimentation with Mobilenet-V2 architecture is presented. The primary contributions of this paper include:

1. The introduction of a modified learning rate policy suited for learning on smaller databases.
2. The provision of a self-created database for sugarcane leaf disease classification.
3. A comparative performance analysis of the modified learning rate policy against established learning rate policies on standard databases: MNIST, CIFAR-10, and the proposed database.

The remainder of the paper is structured as follows: Section 2 reviews related work and provides a succinct overview of available learning rate policies. Section 3 offers a comprehensive explanation of the modified learning rate policy. Section 4 presents the experimental results and discussions. Section 5 concludes the paper, outlining the major findings, limitations, and future research directions.

## 2. RELATED WORK

This section describes the behavior of learning rate schedulers in detail. Plant pathology is a branch of biology that studies plant diseases and the chemicals that cause them. Traditional disease detection technologies require more human interaction and are often less accurate. Deep learning approaches have recently gained a lot of interest in doing such jobs with more precision. Deep learning approaches, on the other hand, necessitate a large enough database to train it. This guarantees that the model is effectively trained and behaves as the user expects. It is not always possible to generate data samples large enough to satisfy the needs of a large database. Many approaches have been taken to address this issue. Image alteration obtained up to 99% accuracy. To alleviate data shortage, more synthetic images based on generative adversarial networks (GAN) have been explored. To leverage the requirement, conditional GAN was used in the research [18] work to generate synthetic diseased cotton-leaf images. The Plant Village database has grown in popularity over the years as the primary option of plant disease researchers. In, around 3500 maize-related photos were separated and used to train the model. Although the Plant Village data-set is well-

known, it primarily contains photos collected in laboratory settings [19]. Selvaraj et al. [20] attempted to get actual photographs from the field. Images were gathered in a variety of environmental conditions in order to create a robust model. This approach preserved sample variety by gathering photos from various places, growth stages, and morphological features. To improve categorization results, a new architecture was used. In the research [21], an efficient tweak to the deep learning architecture demonstrated improved detection of citrus leaf disease. This work has also significantly depended on data augmentation techniques to overcome the scarcity of sufficient datasets. T. absoluta infection on tomato plants was studied more thoroughly in a controlled environment. While photographing plants, special attention was paid to the plant's life cycle. A total of 5235 photos were collected for two classes: healthy and diseased. This emphasizes the difficulty in collecting photos for real-time plant disease classification use cases and prompts a need to invent some strategies that can aid in improving the performance of a small-sized database [22, 23]. Deep learning approaches have evolved in recent years, and many creative changes have attempted to raise the performance of plant disease categorization to a new level. Mohanty et al. used a transfer learning strategy to experiment with the plant village dataset and obtained accuracy of up to 99% using photos collected in a laboratory environment. It should be noted that when images other than similar databases were used for testing, performance deteriorated considerably [24]. To overcome the short data-set constraint, systematically ordered convolutional neural networks (CNNs) were used, and their predictions were aggregated into heat maps before being fed into the final CNN for classification. On test images, an accuracy of around 96.7% was observed [25]. Despite the fact that the photos were obtained from fields, a deep convolutional neural network trained on very small-sized samples performed better. The classification task took into account a total of five classes. This has emphasized the significance of selecting appropriate architecture for deep learning experimentation [26]. A transfer learning architecture processing multi-model inputs was developed in the research [27] to speed up training, reduce the dependency on data, and overcome over-fitting challenges related to small database problems. This multi-fusion model has an accuracy of 98.9% for six tomato plant diseases. Object detection is yet another key technology that accurately locates and identifies diseases and provides significant support with image diagnosis. DF-tiny-YOLO, an enhanced target identification model for apple leaf diseases, was developed in the research [28]. Using it, faster and more effective identification of leaf diseases was reported when dealing with a small set of 1404 images. EfficientNetB0 and DenseNet121 characteristics were merged to achieve higher accuracy in maize leaf disease classification. A slight improvement in model performance was seen when compared to ResNet152 and InceptionV3 [29]. The corn seed dataset was used to test the lightweight network ShuffleNetV2. To begin, Cycle-Consistent Adversarial Networks solved the problem of fewer datasets, while the Efficient Channel Attention module was included to assure increased network performance. To extend the network's receptive field, a modified 77% depth-wise convolution was applied. ShuffleNetV2 repetitions were added to the framework to lighten it. The efficacy of the network structure adjustments in the network was demonstrated by an accuracy of 96.2% and an inference speed of 9.71ms [30]. Attempts were made in the research [31] to process the images properly and employ support vector

machines and k-means. When compared to the convolutional neural networks utilized in the study, effective image modification ensured improved results. In the research [32], hyperspectral and multispectral information processing tools were employed to aid agriculture productivity and practices. The best feasible use of big data, machine learning, and deep learning approaches for agriculture automation was a core research product of the effort. Unmanned aerial vehicle (UAV) images of Sorghum fields with motion blur were used to improve weed management. On a different test set [33], an architecture resembling U-net with a ResNet-34 feature extractor produced an F1-score of nearly 89%. The preceding investigations laid the foundation for dealing with the shortcomings of small database size. Modified network topologies, innovative inclusion, and sometimes image alterations have resulted in considerable performance gains. It has grabbed our interest to investigate the influence of hyperparameter changes on network performance.

The method that controls how the model's weights are updated throughout training is known as a learning rate policy in deep learning. It starts with an initial learning rate that scales the backpropagation gradients. The learning rate may gradually decline over time as part of this policy to fine-tune the model as it gets closer to convergence. A key factor in optimization is the learning rate, which can either cause instability or slow convergence depending on its value. To get the best performance out of deep learning models, the proper learning rate policy and hyperparameters are essential.

A full analysis of learning rate as a hyperparameter is provided below.

### 2.1 Step decay

Step decay is a learning rate scheduler, often known as learning rate annealing. In this process, training begins with a relatively high learning rate and gradually decreases as the model goes through training epochs [34, 35]. Figure 1 (a) shows the behavior of step decay against a number of epochs.

$$lr = base_{lr} * drop^{\lfloor \frac{epoch}{epoch\_drop} \rfloor} \quad (1)$$

In Eq. (1),  $lr$  is the current learning rate,  $base_{lr}$  is the initial learning rate,  $epoch\_drop$  is the factor after which the learning rate drops, and epoch signifies the number of the epoch.

### 2.2 Exponential decay

It is similar to the step decay unlike here the learning rate is decreased exponentially [36, 37]. In Eq. (2),  $lr$  is the current learning rate,  $base_{lr}$  is the initial learning rate,  $k$  is the constant and  $t$  signifies the current epoch. Figure 1 (b) shows the behavior of exponential decay against a number of epochs.

$$lr = base_{lr} * exp^{-kt} \quad (2)$$

### 2.3 Cosine learning rate decay

A cosine learning rate decay decreases the learning rate in such a manner that it forms the shape of the sinusoid. Typically, it is always preferred in restart mode where once the learning rate value reaches a minimum, it is initialized to maximum

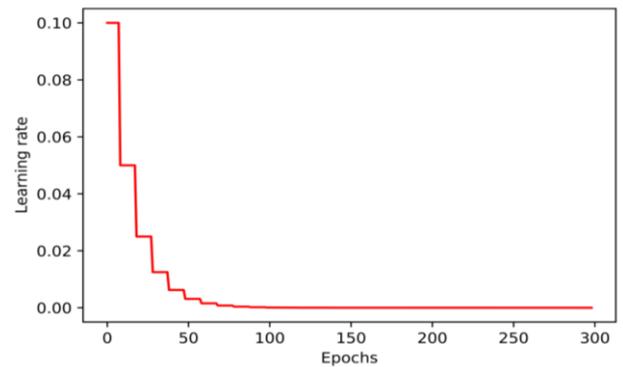
value again, and the cycle repeats until the model converges [38, 39]. In Eq. (3),  $\alpha$  controls the shape of the sinusoid. The response of cosine learning rate decay is shown in Figure 1 (c),

$$lr = base_{lr} * \left( (1 - \alpha) * (0.5 * (1 + \cos(\pi * (\frac{epoch}{Epoch})))) + \alpha \right) \quad (3)$$

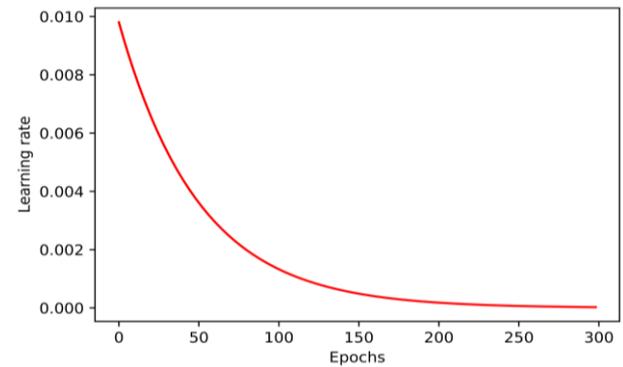
### 2.4 Exponential decay sine

In Figure 1 (d), the decaying and oscillating nature of exponentially decaying sine wave is shown. The current epoch is mentioned by epoch, a total number of epochs is given Epoch.  $b$  is the number of batches per epoch. To control decay and oscillation user can tune  $\alpha$  and  $\beta$ , respectively [40].

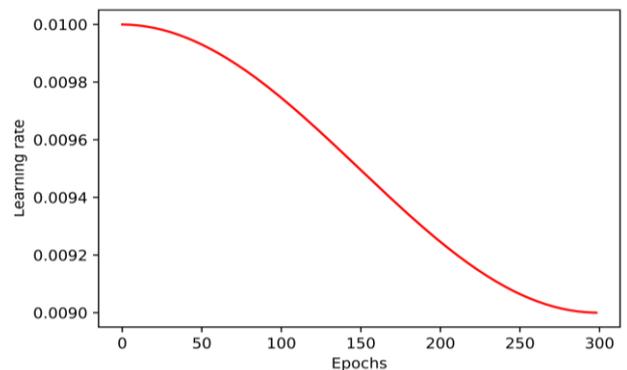
$$lr = base_{lr} * exp\left(\frac{-\alpha * epoch}{Epoch}\right) * \left(\sin\left(\frac{\beta * epoch}{2\pi b}\right) + exp\left(\frac{-\alpha * epoch}{Epoch}\right) + 0.5\right) \quad (4)$$



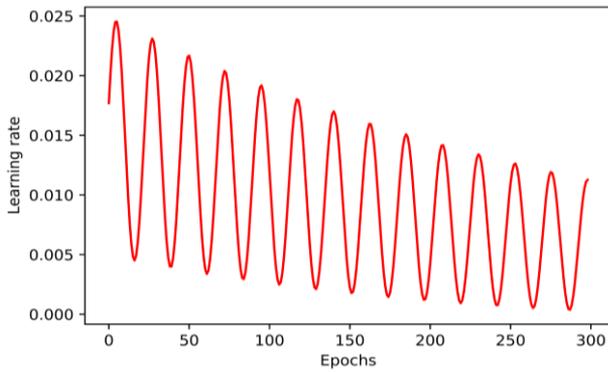
(a) Step



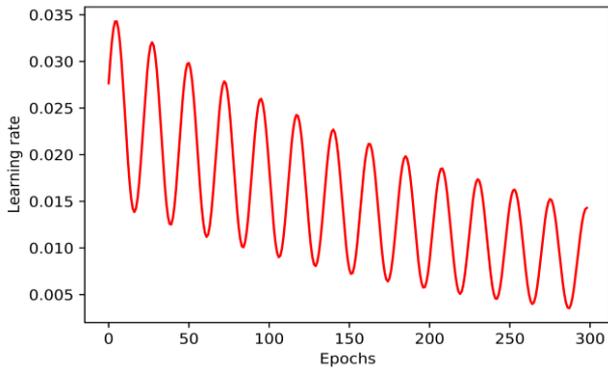
(b) Exponential



(c) Cos



(d) Exponential sine



(e) Proposed

**Figure 1.** Plot for (a) step decay (b) exponential decay (c) cosine decay (d) sine decay (e) proposed

## 2.5 Specificity and contribution

The paper has introduced a whole new self-created database for sugarcane leaf diseases. The reason behind choosing sugarcane for study is its importance in the daily livelihood of Indian farmers. It is one of the major cash crops in India and is heavily impacted due to diseases resulting in financial losses to farmers. In a pursuit to suggest the best methods for disease classification reasons for devising a modified learning rate policy are enlisted in Section 3.1. However, to validate the claims all the methods are tested with MNIST and CIFAR10, popular databases in the machine learning fields.

## 3. PROPOSED METHOD

This section provides a brief discussion of a modified learning rate scheduler, followed by a brief overview of the self-created database and experimentation activity.

### 3.1 Modified learning rate scheduler

An et al. [40] proposed exponential decay sine wave learning for deep neural networks. They presented a unique learning rate scheduler that is simple yet effective, converges faster, and gives satisfactory performance. However, to the best of our knowledge, smaller databases did not fare equally well in terms of classification accuracy.

The contribution of the exponential component in the equation is explored in further depth in the proposed work. In Eq. (5), an additional exponential component was introduced, allowing faster model convergence with the same number of

epochs. In Eq. (5), epoch indicates the iteration number,  $base\_lr$  is the initial learning rate, Epoch means the total number of iterations and  $b$  is several batches per epoch. To control decay and oscillation user can tune  $\alpha$  and  $\beta$ , respectively. Figure 1 (e) clearly shows the impact of the exponential factor added in the learning rate policy.

$$lr = base_{lr} * exp\left(\frac{-\alpha * epoch}{Epoch}\right) * \left(\sin\left(\frac{\beta * epoch}{2\pi b}\right) + exp\left(\frac{-\alpha * epoch}{Epoch}\right) + exp\left(\frac{-\alpha * epoch}{Epoch}\right) + 0.5\right) \quad (5)$$

For any learning rate policy that has been devised, it must satisfy several common criteria as follows:

(1) A good learning rate must be continuous and smooth; otherwise, there will be instability during training.

(2) Monotonicity: It must monotonically decrease.

After adding the extra exponential component, all the necessary and sufficient conditions were fulfilled, resulting in a slight improvement in performance, as indicated in the result tables.

## 3.2 Proposed database

Many humanitarian efforts and material resources were utilized in the early stages of our work to collect diseases on sugarcane leaves because there were few specialized datasets available for the classification of sugarcane leaf diseases. Disease patterns vary according to variation and environment. A total of 2569 photos of sugarcane leaves are obtained, which correspond to five classes: healthy, red rot, mosaic, yellow leaf, and rust. The research work is aimed at the eventual implementation of the real-time system. The photographs of sugarcane leaves were captured using smartphones. These images were later showcased to a group of farmers with rich experience in cultivating sugarcane for more than a decade. To make sure labels are accurate and to ensure quality labels for the images were assigned based on their observations, and these labels were later cross-verified by zonal agricultural officers who have rich experience in plant pathology and the relevant domain. A summary of the database is given in Table 1.

**Table 1.** Images per class in the sugarcane database

Category Name	Number of Images	Percentage
Healthy	520	20.24%
Rust	514	20.07%
Red rot	519	20.20%
Yellow	505	19.65%
Mosaic	511	19.89%
<b>Total</b>	<b>2569</b>	<b>100%</b>

## 4. RESULTS AND DISCUSSIONS

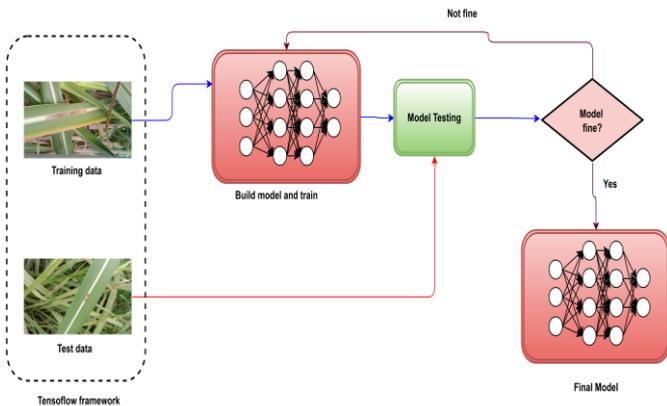
### 4.1 Environment settings

This study takes into account the TensorFlow framework and the Google collaborative environment for training and testing models with various learning rate methods. Python is

the preferred language for this study due to its extensive library support for deep learning approaches. System details are given in Table 2. The overall block diagram of training the neural network is shown in Figure 2.

**Table 2.** System specifications

Parameter	Specification
Language	Python
Compute backend	Google compute engine backend (GPU)
GPU	Tesla T4
<b>Cuda cores</b>	<b>2560</b>
Memory interface	256bit
Memory type	GDDR6
Bandwidth	900GB/s
GPU RAM	16GB



**Figure 2.** Overall block diagram of methodology

#### 4.2 Database description

Two well-known benchmark databases, MNIST [41] and CIFAR-10 [42], are explored in this paper, coupled with one self-created database for sugarcane disease classification. The MNIST dataset contains 70000 gray-scale images of handwritten digits. Each image is 28×28 in size. Here, 60000 images are for training and the rest are for testing. This dataset has ten classes of images that range from 0 to 9. Cifar-10 comprises ten classes, such as animals, birds, and airplanes. Every class has 6000, 32×32 colored images. A total of 60000 images are distributed in an 80:20 ratio for training and testing. Experimentation is carried out on these datasets in order to evaluate the various learning rate schedules. Table 3 presents statistical data on the databases used in the study. The proposed dataset is divided into five categories: healthy, rust, yellow leaf, red rot, and mosaic. The database is well-balanced, with approximately 500 images in each class. Images in each class are increased fivefold after the augmentation.

**Table 3.** Database description

Database	Classes	Number of Images	Image Size
MNIST	10	60000	28×28
CIFAR-10	10	10000	32×32
Proposed	5	2569	128×128
Proposed (Augmentation)	5	12845	128×128

#### 4.3 Data augmentation

It is recommended practice to augment the database for deep neural network training (DNN). When small databases are utilized for learning, irrelevant features have a negative impact on model learning. DNN performance cannot be affected by translation, size, illumination, or viewpoint. Image processing procedures such as flip, rotation, noise addition, and translation can all help to expand the database size. This aids in preventing the model under training from being overfitting. Manual image collection is a time-consuming and difficult task. All learning rates and databases are evaluated using the cross-validation technique. A total of 10 subsets again in the same fraction of ‘8: 1: 1’ is used as “training: validation: testing”. The average classification results are reported in this work. Data augmentation has helped to overcome shortcomings like overfitting that were observed initially due to small databases. The data augmentation details are mentioned as follows.

#### 4.4 Architecture and performance parameters

Several experiments were carried out in order to select the best architecture. The major idea was to test the performance of architecture with small databases. A few good architectures were investigated, and the Mobilenet-V2 provided the highest possible accuracy. Table 4 demonstrates that mobilenet-V2 has the highest accuracy of all for small databases. This has helped in the selection of Mobilenet-V2 as the backbone architecture or the subsequent testing associated with the suggested learning rate policies. These backbones were used with pretrained ImageNet weights. During training, the bottom layers were frozen, while the top layers remained adaptable to the training data. The care was taken to keep the hyperparameters almost equal during the experimentation. These values are selected based on the previous experimentation that researcher performed which is available in the research [43].

Due to its ability to take into consideration true positives, true negatives, false positives, and false negatives, F1 score, precision, recall, and accuracy are well suited for image classification and offer an in-depth evaluation of model performance. As because of their lack of sensitivity that allows for subtle classification differences, regression-focused measures like mean squared error (MSE) and R-squared are inadequate for classifying images.

Precision:

It is the ratio of retrieved and relevant results to all retrieved results. It is the description of the ratio between true positive to the total positive results obtained during the experimentation, as shown in Eq. (6).

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

*Recall:* Recall sometimes referred to as sensitivity is the ratio of retrieved and relevant results to the relevant results. In Eq. (7), *TP* is correct detection, *FP*, is the total number of false detections and *FN* is several missed quantities.

**Table 4.** Data augmentation details

Class	Actual	Flip	Rotation (180)	Noise (Gaussian)	Translation (Down)	Total
Healthy	520	520	520	520	520	2600
Rust	514	514	514	514	514	2570
Red rot	519	519	519	519	519	2595
Yellow	505	505	505	505	505	2525
Mosaic	511	511	511	511	511	2555
<b>Total</b>	<b>2569</b>	<b>2569</b>	<b>2569</b>	<b>2569</b>	<b>2569</b>	<b>12845</b>

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

*F1-Score:* The F1 score is also referred to as the F measure. It signifies the harmony between precision and recall and helps the researcher to trade-off between them. Mathematically, it is given by Eq. (8).

$$F1 - score = \frac{2 * TP}{2 * TP + FP + FN} \tag{8}$$

*Accuracy:* It is the ratio of several correct identifications to all identifications in the experimentation and is given by Eq. (9).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

**Table 5.** Performance comparison of different models

Model	Accuracy	Number of Parameters
VGG19	0.70	20,026,948
ResNet50	0.81	23,575,044
XceptionNet	0.79	20,871,724
<b>MobileNetV2</b>	<b>0.83</b>	<b>2,264,388</b>
EfficientNet_B7	0.73	2,264,388

**Table 6.** Hyperparameters

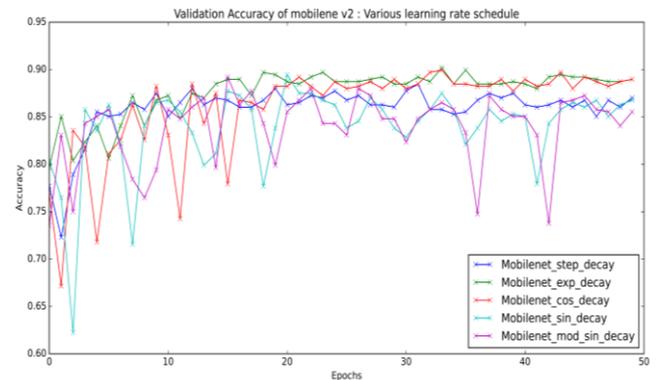
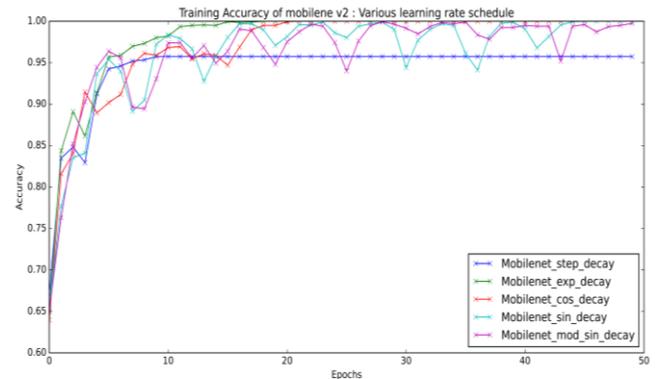
Hyperparameter	Typical Value
Learning Rate	0.003
Batch Size	32
Number of Epochs	50
Optimizer	Adam
Weight Decay (L2 Norm)	0.001
Input Image Size	As per architecture
Number of Top Layers	Same for all

During the experimentation, it was discovered that the proposed learning rate policy accelerated the training process. The learning rate is expected to be higher during the beginning period of training and to decrease towards the end. However, in sine decay (Figure 1 (d)), the learning rate oscillates more than necessary during each cycle, increasing training time. However, the modified learning rate policy Figure 1 (e) has limited the extreme movement of LR, increasing the speed with higher classification performance. Figure 3 depicts the training and validation curves for the 50 epochs with the sugarcane leaf database used in the paper. Experiments with the MNIST and CIFAR datasets yielded the greatest F1 scores of 99% and 92%, respectively with the proposed LR. The overall F1-score for the own database is comparatively low with respect to the standard databases, however, the learning

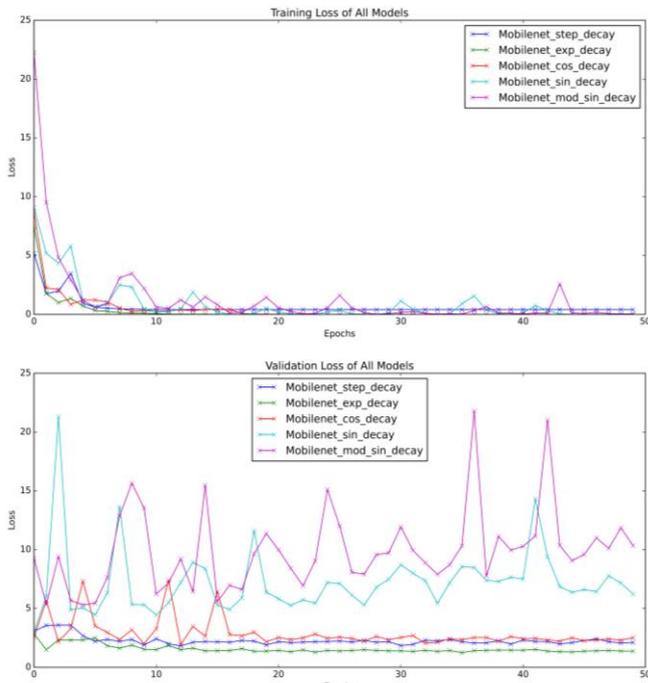
rate policy proposed in the experimentation has shown slight improvement in it with respect to traditional learning rate policies. During the testing phase, classification performance is shown in Figure 4 as being more accurate.

**Table 7.** Mobilenet-V2 with various learning rate scheduler

	Learning Rate Policy	Precision	Recall	F1-Score	Accuracy
MNIST	step	0.98	0.98	0.99	0.99
	exponential	0.99	1	1	0.99
	cosine	0.99	0.99	0.99	0.99
	exponential sine	0.96	0.96	0.94	0.95
	proposed	0.98	0.98	0.99	0.98
CIFAR10	step	0.84	0.84	0.84	0.87
	exponential	0.79	0.79	0.79	0.78
	cosine	0.88	0.9	0.91	0.9
	exponential sine	0.89	0.91	0.92	0.92
	proposed	0.92	0.93	0.92	0.92
Proposed database	step	0.82	0.78	0.78	0.78
	exponential	0.86	0.87	0.86	0.87
	cosine	0.84	0.82	0.83	0.84
	exponential sine	0.86	0.85	0.85	0.85
	proposed	0.88	0.89	0.89	0.87



(a) Accuracy curves



(b) Loss curves

**Figure 3.** Training and validation plots with proposed database for all learning rates



**Figure 4.** Classification results with proposed database for proposed learning rate

Table 5 gives the performance comparison of all models and Table 6 highlights the hyperparameters used in the experimentation. Although the combined comparison of all learning rate policies for training and validation performance reveals greater oscillations in the suggested learning rate, Table 7 clearly suggests improved performance in F1-score for all databases used in experimentation. Proposed scheduler has slightly improved performance, as evident from the results. The addition of an exponential term has enabled faster convergence and constrained the gradient from deviating unnecessarily towards extreme ends compared to the typical learning rate scheduler's response.

## 5. CONCLUSION

The findings in this research highlight the advantages of incorporating an exponential term into the learning rate equation. To the best of our knowledge, the decision to choose Mobilenet-V2 architecture was based on empirical results achieved with several architectures. This work is done with the help of a small database. For smaller datasets, the proposed experiment demonstrated that the proposed learning rate policy performs better than existing strategies. The recommended learning rate offers improved classification accuracies, but it also has more oscillations. In future, we intend to minimise these oscillations to an acceptable level with improved classification performance. For improved results, we intend to expand the number of images in the dataset. Furthermore, we believe that the performance of the proposed learning rate with efficient novel architecture will boost the overall performance.

## REFERENCES

- [1] Bishop, C.M., Nasrabadi, N.M. (2006). Pattern recognition and machine learning. New York: Springer, 4(4): 738.
- [2] Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. arXiv Preprint arXiv: 1207.0580. <https://doi.org/10.48550/arXiv.1207.0580>
- [3] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT Press. <http://www.deeplearningbook.org>.
- [4] Amari, S.I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2): 251-276. <https://doi.org/10.1162/089976698300017746>
- [5] Ge, R., Kakade, S.M., Kidambi, R., Netrapalli, P. (2019). The step decay schedule: A near optimal, geometrically decaying learning rate procedure for least squares. *Advances in Neural Information Processing Systems*, 32.
- [6] Pascanu, R., Bengio, Y. (2013). Revisiting natural gradient for deep networks. arXiv Preprint arXiv: 1301.3584. <https://doi.org/10.48550/arXiv.1301.3584>
- [7] Forst, W., Hoffmann, D. (2010). Optimization-theory and practice. Springer Science & Business Media.
- [8] Schraudolph, N.N., Graepel, T. (2002). Towards stochastic conjugate gradient methods. In *Proceedings of the 9th International Conference on Neural Information Processing*, Singapore, pp. 853-856. <https://doi.org/10.1109/ICONIP.2002.1198180>
- [9] Llugsí, R., El Yacoubi, S., Fontaine, A., Lupera, P. (2021). Comparison between Adam, AdaMax and Adam W optimizers to implement a Weather Forecast based on Neural Networks for the Andean city of Quito. In *2021 IEEE Fifth Ecuador Technical Chapters Meeting (ETCM)*, Cuenca, Ecuador, pp. 1-6. <https://doi.org/10.1109/ETCM53643.2021.9590681>
- [10] Zhang, N., Lei, D., Zhao, J.F. (2018). An improved Adagrad gradient descent optimization algorithm. In *2018 Chinese Automation Congress (CAC)*, Xi'an, China, pp. 2359-2362. <https://doi.org/10.1109/CAC.2018.8623271>
- [11] Indrapriyadarsini, S., Mahboubi, S., Ninomiya, H., Asai, H. (2018). Implementation of a modified Nesterov's Accelerated quasi-Newton Method on Tensorflow. In

- 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Orlando, FL, USA, pp. 1147-1154. <https://doi.org/10.1109/ICMLA.2018.00185>
- [12] Zeiler, M.D. (2012). Adadelata: An adaptive learning rate method. arXiv Preprint arXiv: 1212.5701. <https://doi.org/10.48550/arXiv.1212.5701>
- [13] Wen, L., Li, X., Gao, L. (2020). A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification. *IEEE Transactions on Industrial Electronics*, 68(12): 12890-12900. <https://doi.org/10.1109/TIE.2020.3044808>
- [14] Iiduka, H. (2021). Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks. *IEEE Transactions on Cybernetics*, 52(12): 13250-13261. <https://doi.org/10.1109/TCYB.2021.3107415>
- [15] Gulde, R., Tuscher, M., Csiszar, A., Riedel, O., Verl, A. (2020). Deep reinforcement learning using cyclical learning rates. In 2020 Third International Conference on Artificial Intelligence for Industries (AI4I), Irvine, CA, USA, pp. 32-35. <https://doi.org/10.1109/AI4I49448.2020.00014>
- [16] Bulut, B., BÜTÜN, E., Kaya, M. (2022). Polyp segmentation in colonoscopy images using U-Net and cyclic learning rate. In 2022 International Conference on Decision Aid Sciences and Applications (DASA), Chiangrai, Thailand, pp. 1149-1152. <https://doi.org/10.1109/DASA54658.2022.9765101>
- [17] Nie, Y., Carratù, M., O'Nils, M., Sommella, P., Moise, A.U., Lundgren, J. (2022). Skin cancer classification based on cosine cyclical learning rate with deep learning. In 2022 IEEE International Instrumentation and Measurement Technology Conference (I2MTC), Ottawa, ON, Canada, pp. 1-6. <https://doi.org/10.1109/I2MTC48687.2022.9806568>
- [18] Ramanadham, K.L., Savarimuthu, N. (2022). vCrop: An automated plant disease prediction using deep ensemble framework using real field images. *Sādhanā*, 47(4): 268. <https://doi.org/10.1007/s12046-022-02041-8>
- [19] Liu, Y., Gao, G., Zhang, Z. (2022). Crop disease recognition based on modified light-weight CNN with attention mechanism. *IEEE Access*, 10: 112066-112075. <https://doi.org/10.1109/ACCESS.2022.3216285>
- [20] Selvaraj, M.G., Vergara, A., Ruiz, H., Safari, N., Elayabalan, S., Ocimati, W., Blomme, G. (2019). AI-powered banana diseases and pest detection. *Plant Methods*, 15: 1-11. <https://doi.org/10.1186/s13007-019-0475-z>
- [21] Hassam, M., Khan, M.A., Armghan, A., Althubiti, S.A., Alhaisoni, M., Alqahtani, A., Kadry, S., Kim, Y. (2022). A single stream modified mobilenet V2 and whale controlled entropy based optimization framework for citrus fruit diseases recognition. *IEEE Access*, 10: 91828-91839. <https://doi.org/10.1109/ACCESS.2022.3201338>
- [22] Loyani, L.K., Bradshaw, K., Machuve, D. (2021). Segmentation of tuta absoluta's damage on tomato plants: A computer vision approach. *Applied Artificial Intelligence*, 35(14): 1107-1127. <https://doi.org/10.1080/08839514.2021.1972254>
- [23] Shamalik, R., Koli, S. (2022). DeepHands: Dynamic hand gesture detection with depth estimation and 3D reconstruction from monocular RGB data. *Sādhanā*, 47(4): 247. <https://doi.org/10.1007/s12046-022-02026-7>
- [24] Mohanty, S.P., Hughes, D.P., Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7: 1419. <https://doi.org/10.3389/fpls.2016.01419>
- [25] DeChant, C., Wiesner-Hanks, T., Chen, S., Stewart, E.L., Yosinski, J., Gore, M.A., Nelson, R.J., Lipson, H. (2017). Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology*, 107(11): 1426-1432. <https://doi.org/10.1094/PHYTO-11-16-0417-R>
- [26] Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., Sun, Z. (2018). A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. *Computers and Electronics in Agriculture*, 154: 18-24. <https://doi.org/10.1016/j.compag.2018.08.048>
- [27] Zhang, N., Wu, H., Zhu, H., Deng, Y., Han, X. (2022). Tomato disease classification and identification method based on multimodal fusion deep learning. *Agriculture*, 12(12): 2014. <https://doi.org/10.3390/agriculture12122014>
- [28] Di, J., Li, Q. (2022). A method of detecting apple leaf diseases based on improved convolutional neural network. *Plos One*, 17(2): e0262629. <https://doi.org/10.1371/journal.pone.0262629>
- [29] Amin, H., Darwish, A., Hassanien, A.E., Soliman, M. (2022). End-to-end deep learning model for corn leaf disease classification. *IEEE Access*, 10: 31103-31115. <https://doi.org/10.1109/ACCESS.2022.3159678>
- [30] Lu, L., Liu, W., Yang, W., Zhao, M., Jiang, T. (2022). Lightweight corn seed disease identification method based on improved shufflenetv2. *Agriculture*, 12(11): 1929. <https://doi.org/10.3390/agriculture12111929>
- [31] Javidan, S.M., Banakar, A., Vakilian, K.A., Ampatzidis, Y. (2023). Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning. *Smart Agricultural Technology*, 3: 100081. <https://doi.org/10.1016/j.atech.2022.100081>
- [32] Ang, K.L.M., Seng, J.K.P. (2021). Big data and machine learning with hyperspectral information in agriculture. *IEEE Access*, 9: 36699-36718. <https://doi.org/10.1109/ACCESS.2021.3051196>
- [33] Genze, N., Ajekwe, R., Güreli, Z., Haselbeck, F., Grieb, M., Grimm, D.G. (2022). Deep learning-based early weed segmentation using motion blurred UAV images of sorghum fields. *Computers and Electronics in Agriculture*, 202: 107388. <https://doi.org/10.1016/j.compag.2022.107388>
- [34] 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 (SCEECS), Bhopal, India, pp. 1-5. <https://doi.org/10.1109/SCEECS48394.2020.94>
- [35] Vinay, B.N., Shah, P.J., Shekar, V., Vanamala, H.R. (2020). Detection of melanoma using deep learning techniques. In 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, pp. 391-394. <https://doi.org/10.1109/ICCAKM46823.2020.9051495>
- [36] Mishra, P., Sarawadekar, K. (2019). Polynomial learning rate policy with warm restart for deep neural network. In

- TENCON 2019-2019 IEEE Region 10 Conference (TENCON), Kochi, India, pp. 2087-2092. <https://doi.org/10.1109/TENCON.2019.8929465>
- [37] Zhu, W., Tang, Y. (2021). Dalu: Adaptive learning rate update in distributed deep learning. In 2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI), Atlanta, GA, USA, pp. 203-209. <https://doi.org/10.1109/SWC50871.2021.00036>
- [38] Mori, M., Nakano, M. (2018). Efficient cyclic learning rate schedules and their evaluations for neural network ensemble. In 2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP), Aalborg, Denmark, pp. 1-6. <https://doi.org/10.1109/MLSP.2018.8517060>
- [39] Lv, J.C., Yi, Z., Tan, K.K. (2007). Global convergence of GHA learning algorithm with nonzero-approaching adaptive learning rates. IEEE Transactions on Neural Networks, 18(6): 1557-1571. <https://doi.org/10.1109/TNN.2007.895824>
- [40] An, W., Wang, H., Zhang, Y., Dai, Q. (2017). Exponential decay sine wave learning rate for fast deep neural network training. In 2017 IEEE Visual Communications and Image Processing (VCIP), St. Petersburg, FL, USA, pp. 1-4. <https://doi.org/10.1109/VCIP.2017.8305126>
- [41] Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Signal Processing Magazine, 29(6): 141-142. <https://doi.org/10.1109/MSP.2012.2211477>
- [42] Krizhevsky, A., Nair, V., Hinton, G. (2018). Cifar-10. Canadian Institute for Advanced Research.
- [43] Daphal, S.D., Koli, S.M. (2023). Enhancing sugarcane disease classification with ensemble deep learning: A comparative study with transfer learning techniques. Heliyon, 9(8).