[1]Adityawardhan Mishra
[1]Manivel Kandasamy
[1]Vansh Tiwari
[1]Arpit Sharma
[1]Rohan Patil
[1]Vedant Dwivedi

Optimizing Agriculture Resilience: Transfer Learning and Advanced Optimizers in Plant Disease Detection

Abstract: - In recent decades, the global population surge has heightened the demand for food, placing agricultural produce at risk of various diseases that can compromise yields. Addressing this challenge, early detection of plant diseases becomes imperative to prevent their spread throughout crops. Plant leaves, being the first part of the plant body to exhibit abnormalities, serve as crucial indicators for tracking diseases. Over the past decade, significant efforts have been invested in leveraging the PlantVillage Dataset to develop effective methodologies for disease detection. This study focuses on employing two transfer learning models, ConvNeXtV2 and ViT, along with a novel optimization technique called LION. The objective was to evaluate the efficacy of this approach in training models and assessing the final results through various performance metrics on the validation set. The findings underscore the robustness of the Lion optimizer in facilitating efficient learning and refinement of representations for both ViT and ConvNeXtV2 models. The best accuracy achieved, a remarkable 99.48%, highlights the effectiveness of this optimization technique. In conclusion, our study suggests that transformer-based models, such as ViT, outperform earlier model structures in the context of plant disease detection. The LION optimizer proves instrumental in achieving faster convergence and higher accuracy, further enhancing the overall performance of transformer-based architectures.

Keywords: Transfer Learning, Advanced Optimizer, Plant Disease Detection, Neural Networks, Transformers

I. INTRODUCTION

The worldwide agricultural landscape is facing an impending and rising threat in the shape of widespread plant diseases, which pose a significant challenge to the critical pillar of food security. The importance of early, precise identification of these illnesses is highlighted by the need for prompt action to establish effective mitigation methods. In this context, recent years have seen a transformational wave in agricultural technology, with the incorporation of sophisticated methodologies, particularly transfer learning and optimized deep learning models, serving as a beacon of hope for changing the area of plant disease identification.

In an era where traditional illness detection approaches are frequently inaccurate and inefficient, the use of transfer learning emerges as a critical option. This strategy involves pre-training neural networks on large datasets, giving them the ability to exploit gained information from other fields. The use of ConvNext and ViT, due to their effectiveness in detecting detailed patterns in pictures, represents a deliberate attempt to lay a solid basis for the nuanced task of plant disease diagnosis.

In this pursuit, the AdamW and Lion optimizers take the lead, meticulously fine-tuning the both ConvNext and ViT model. AdamW not only seeks to enhance accuracy but also to strengthen the model's flexibility in the face of complicated agricultural datasets. This resilience is especially important when dealing with the inherent complexity and variety of plant diseases in agricultural datasets. The objective here is twofold: to improve accuracy and convergence speed, making these models more exact and efficient for real-world agricultural applications. LION optimization technique aims not just to accelerate convergence, but also to improve the model's overall performance.

This study addresses the urgent need for better disease detection technologies, which makes a significant contribution to the conversation on sustainable agriculture. Through an examination of the interplay between transfer learning and optimal deep learning models, this study provides valuable insights for agricultural stakeholders who are aiming to achieve global food security. The study, which is at the intersection of technology and agriculture, advances our understanding of plant diseases and encourages resilience and sustainability in global food systems. This study opens the door to a new era where data-driven insights enable stakeholders to make informed decisions as we navigate the intersection of agricultural resilience and technological innovation. The

[1] Affiliation: Unitedworld Institute of Technology, Karnavati University, Gandhinagar, Gujarat, India-382422

techniques, datasets, and intricacies of optimization processes are explored in later sections, highlighting the difficulties involved in this revolutionary approach to plant disease identification.

II. LITERATURE REVIEW

This section discusses previous works done with similar motivation on the PlantVillage Data corpus for the purposes of plant disease detection. Yafeng Zhao et al.[1] propose the use of DoubleGAN to generate high-resolution images of unhealthy plant leaves to balance unbalanced datasets . The DoubleGAN consists of two stages: in stage 1, healthy and unhealthy leaf images are used as inputs to train a Wasserstein generative adversarial network (WGAN) to obtain a pretrained model. This pretrained model is then used to generate 64x64 pixel images of unhealthy leaves. In stage 2, a super-resolution generative adversarial network (SRGAN) is used to obtain corresponding 256x256 pixel images to expand the unbalanced dataset. The generated images using DoubleGAN are clearer than those generated by DCGAN (Deep convolution generative adversarial network), and the accuracy of plant species and disease recognition reached 99.80 and 99.53, respectively, which is better than the original dataset.

Edna Chebet Too et al.[2] focus on fine-tuning and evaluating deep convolutional neural network architectures for image-based plant disease classification. The architectures evaluated include VGG 16, Inception V4, ResNet with 50, 101, and 152 layers, and DenseNets with 121 layers. DenseNets showed consistent improvement in accuracy with growing epochs and achieved a testing accuracy score of 99.75, outperforming other architectures. The study aims to develop fast and accurate models for plant disease identification to address food security concerns.

Sk. Mahmudul Hassan et al. [3] leverage diverse CNN architectures, including InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNetB0, trained on a comprehensive dataset featuring 14 plant species and 38 disease classes. The implemented models demonstrate superior performance in disease classification, surpassing traditional handcrafted-feature-based methods and other deep-learning models in terms of accuracy and training time. Notably, the MobileNetV2 architecture ensures compatibility with mobile devices, suggesting the potential for real-time deployment in agricultural systems. Overall, the deep CNN model exhibits promise for efficient and accurate plant disease identification, showcasing its viability for practical applications in agriculture.

Bincy Chellapandi et al. [4] focus on using deep learning-based models such as VGG16, VGG19, ResNet50, InceptionV3, InceptionResnetV2, MobileNet, MobileNetV2, DenseNet and transfer learning to classify images of diseased plant leaves, achieving the best result with an accuracy of 99% using the DenseNet model.

Aydin Kaya et al. [5] used Deep learning models, trained over 100 epochs, underwent feature learning experiments using linear kernel Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA) classifiers. Observations highlight the impact of dataset size on CNN model performance, emphasizing the general principle that more data leads to improved classification outcomes. Cross-validation parameters were chosen considering the trade-off between accuracy and computational cost.

Mobeen Ahmad et al. [6] propose an efficient method for classifying plant disease symptoms using memory-efficient convolutional neural networks, reducing training times and optimizing for resource-constrained devices. It also addresses the class imbalance problem and introduces a stepwise transfer learning approach to prevent negative transfer learning. The proposed system achieves high accuracy on both the PlantVillage and pepper disease datasets, outperforming previous works, with 99% accuracy on the Pepper dataset and 99.69% accuracy on the PlantVillage dataset.

S. Pudumalar et al. [7] introduce, the Hydra framework, an ensemble deep learning model, for symptom-wise recognition of cotton diseases, achieving an accuracy of 95% in disease recognition. The Hydra framework combines Convolutional Neural Network (CNN) and VGG16 model with SoftMax function and ReLU activation to improve the performance of disease detection in cotton crops.

Yasamin Borhani et al. [8] used the Vision Transformer (ViT) for automated plant disease classification, aiming to provide visual information to farmers for preventive measures. The ViT structure follows the human approach of focusing on specific areas of an image for classification. The study compares the performance of ViT, classical CNN methods, and a combination of CNN and ViT for plant disease classification. It is concluded that attention blocks in ViT increase accuracy but slow down prediction, while combining attention blocks with CNN blocks can

compensate for speed. Early diagnosis of plant disease is crucial, and the use of artificial intelligence can aid in accurate recognition. The paper also discusses the use of pre-designed architectures like ResNet and ViT for computer vision tasks, highlighting the need for large datasets for optimal parameter values.

R. S. Sandhya Devi et al. [9] used the EfficientNetV2 model for accurately classifying plant diseases and recognizing pests. The model is trained on unbalanced multi-class datasets and utilizes techniques such as Mixup Augmentation and regularization to improve performance. The evaluation metrics used in the study are discussed, and the results of the EfficientNetV2 classifier are presented.

III. METHODOLOGY

A. Image Corpus

A version [10] of the Plant Village dataset, the Plant Village dataset consists of 54,305 high-resolution images that illustrate 38 different plant disease classes. It is a useful tool for agricultural research and the study of plant diseases. To improve dataset diversity and boost model robustness, image augmentation techniques like Color Jitter [11], Random Rotation [12], and Random Vertical Flip [13] were utilized. Random Rotation adds variability by rotating images within predetermined degrees, Random Vertical Flip improves dataset diversity by flipping images vertically, and Color Jitter introduces controlled variations in color distributions. These methods help to train machine learning models that are more robust and adaptive, especially when it comes to tasks like object detection and image classification.

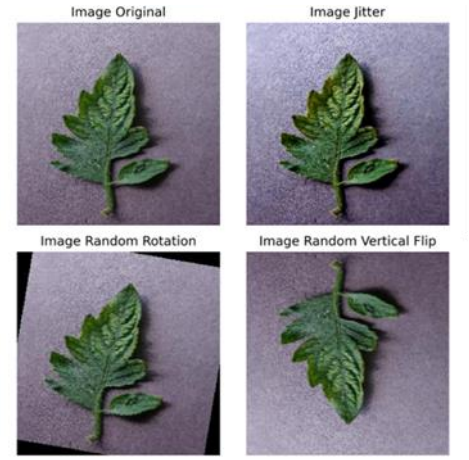

Figure 1: Demonstration of Image Augmentation on Mosaic-Virus Tomato Image

Table 1: Original Plant Village Image Corpus

Plant	**Disease**
Apple (3172)	Gymnosporangium juniper-virginianae (275), Venturia insequalis (630), Botryospaeria obtuse (621), Healthy (1645)
Blueberry (1502)	Healthy (1502)
Cherry (1906)	Podosphaera spp (1052), Healthy (854)
Corn (3852)	Cercospora zeae-maydis (513), Puccinia sorghi (1192), Exserohilum turcicum (985), Healthy (1162)

Grape (4063)	Guignardia bidwellii (1180), Phaeomoniella spp. (1383), Pseudocercospor a ViTis (1076), Healthy (423)
Orange (5507)	Candidatus Liber ibacter (5507)
Peach (2657)	Xanthomonas campestris (2297), Healthy (360)
Bell Pepper (2475)	Xanthomonas campestris(997), Healthy (1478)
Potato (2152)	Alternaria solani (1000), Phytophthora Infestans(1000), Healthy(152)
Raspberry (371)	Healthy (371)
Soybean (5090)	Healthy (5090)
Squash (1835)	Sphaerotheca fuliginea (1835)
Strawberry (1565)	Diplocarpon earlianum (1109), Healthy (456)
Tomato (18,160)	Alternaria solani (1000), Septoria lycopersici (1771), Corynespora cassiicola (1404), Fulvia fulva (952), Xanthomonas campestris pv. Vesicatoria (2127), Phytophthora Infestans (1909),Tomato Yellow Leaf Curl Virus (5357), Tomato Mosaic Virus (373), Tetranychus urticae (1676), Healthy (1591)

Table 2 Post-Augmentation class size on Plant Village Image Corpus

Class	**Before Augmentation**	**After Augmentation**
Apple Scab (Venturia insequalis)	630	1260
Black Rot (Botryospaeria obtuse)	621	1242
Cedar Apple Rust		
(Gymnosporangium juniper-virginianae)	275	550
Leaf Mold (Fulvia fulva)	952	1904
Tomato Mosaic Virus	373	746
Healthy	152	304
Early Blight (Alternaria solani)	1000	2000
Late Blight (Phytophthora Infestans)	1000	2000
Gray leaf spot (Cercospora zeae-maydis)	513	1026
Northern Leaf Blight (Exserohilum turcicum)	985	1970
Bacterial Spot (Xanthomonas campestris)	997	1994

Healthy	423	846
Healthy	360	720
Healthy	371	742
Healthy	456	912
Healthy	854	1708

B. Pre-Trained Model

1) ConvNextV2

Convolutional Neural Network eXtension, or ConvNeXt [14], is a family of powerful image classification models that excels on several benchmark datasets due to its scalability. ConvNeXt v2[15] raises the standard for image classification performance and efficiency with its sophisticated architectural decisions and dynamic adaption methods. In exchange for Dynamic Activation Scale (DAS), it abandons Layer Scale Decay [16] and adjusts scaling to individual activations inside each layer in the hopes of improving performance. Feature maps are sharpened and information flow is improved with improved down-sampling using smaller kernels and more residual connections. Modules for attention augmentation periodically provide context and long-range dependencies, increasing representational strength.

Normalizing neural network activations across all channels is accomplished by the use of Global Response Normalization [15], or GRN. For a convolutional or fully connected layer, GRN is frequently applied to its outputs. The mean and standard deviation of each channel activation are first calculated in order for it to function. Each activation is then divided by the standard deviation to determine the mean. Thus, a normalized activation map is present for every channel. Gaussian Error Linear Unit [17], or GeLU, is a kind of activation function. GeLU, a smooth approximation of the rectified linear unit (ReLU) [18], is increasing adoption because of its beneficial features, which include smoothness and effective deep neural network training. In conclusion, Adaptive Residual Drop Path serves as a substitute for Stochastic Depth Drop Path. It offers adjustable regularization by dynamically removing individual residual connections rather than entire layers. ConvNeXt v2 is now a cutting-edge image classification powerhouse because of these improvements, which also result in increased accuracy, efficiency, and generalizability.

It is anticipated that ConvNeXt V2's base model design would reduce dimensionality while maintaining significant input data characteristics. Accuracy is increased and the training process is stabilised by the residual connections and layer normalisation. This model is composed of 4 stages with different blocks, each with distinct input and output characteristics. A down sampling layer in each stage reduces a signal's or dataset's sample count.

The base architecture of each block includes the following layers: conv2d, pwconv1, pwconv2 layers and identity, with some activation functions. The depth wise convolution layer, or dconv (conv2d), is probably employed for dimensionality reduction. Layer normalization is used to normalize the activations across all channels to aid in stabilising the training process. The linear pointwise convolution layer pwconv1 is followed by a GRN and a GeLU activation. Gaussian error linear unit (GeLU) activation function is a smooth gradient and Gaussian response normalization (GRN) which aids in normalizing the activations across various spatial locations. Another linear pointwise convolution layer pwconv2 is used. Identity transfers the second pwconv2 layer's output straight to the residual block's input, bypassing it. Information that may be lost due to convolutions is preserved with the use of this skip link.

ConvNeXt V2 Stage0 has three identical layers of Block 1, followed by ConvNeXt V2 Stage1 which consists of three identical layers of Block 2, ConvNeXt V2 Stage2 which consists of 27 identical layers of Block 3, and ConvNeXt V2 Stage3 which consists of three identical layers of Block 4. There are four stages, as was previously described, and each stage is connected to the others in order to form the 36-layer model.

The FCMAE framework [15] is used to pretrain ConvNeXt V2 models, and the ImageNet-1K [19] dataset at 224x224 resolution is used to refine them. The FCMAE framework significantly enhances the performance of

ConvNeXt models on several recognition benchmarks by combining architectural enhancements with self-supervised learning strategies. The model's performance is further improved by fine-tuning on the ImageNet-1K dataset, which helps it adapt to certain visual identification tasks. ConvNeXt V2 models may achieve state-of-the-art accuracy on the ImageNet-1K dataset using publically accessible data by combining the FCMAE framework with fine-tuning on ImageNet-1K.

2) ViT (Vision Transformers)

Visual Transformers [20] (ViT) offer a paradigm change in computer vision by replacing traditional convolutional layers with self-attention techniques. ViT has exhibited outstanding performance across a wide range of tasks, promoting advances in image recognition and feature learning among the scientific community. An image is selected and fixed-size patches of divisions are made of it, which are then linearly embedded, position embeddings added, and the subsequent sequence of vector is then fed into a conventional Transformer [32] encoder. The usual strategy of inserting an extra learnable "classification token" into the sequence is utilized to perform classification. The traditional approach where, convolutional neural networks (CNNs) are commonly used to process images since they rely on filters to extract characteristics from particular portions of the image. CNNs are excellent at capturing spatial relationships inside images, but they struggle with long-term dependencies and global context. A transformer is a model in deep learning which utilizes mechanisms of attention to weigh the significances of each segment of the incoming data sequences differentially.

Transformers are made up of many self-attention layers which are largely employed in Computer Vision (CV)[21], Natural Language Processing (NLP)[22] and Artificial Intelligence (AI). The self-attention mechanism is an important component of the transformer architecture, which is used to extract long-term dependencies and contextual information from incoming data. The self-attention mechanism allows a ViT model to prioritize relevant input data for the task at hand.

ImageNet1K [19] is a subset of the broader ImageNet dataset, commonly also known as ILSVRC 2012. It's frequently used to train deep learning models for computer vision tasks. Some important facts about the ImageNet1K [19] dataset is that it spams 1000 object classes and there are 1,281,167 training images, 50,000 validation images, and 100,000 test images in it, also the photos in the collection are arranged in a WordNet hierarchy. A "synonym set" or "synset" is an important concept in WordNet that may be defined by a large number of words or phrases. ImageNet aspires to supply 1000 photographs to illustrate each synset on average.

C. Optimizer

1) AdamW

AdamW [23] is a well-known neural network optimizer that combines the Adam and Weight Decay [24] techniques. Adaptive Moment Estimation, or Adam for short, quickly optimizes neural network weights and biases by taking advantage of momentum. It updates current gradients by computing exponentially decaying averages of past gradients, and modifies learning rates for each parameter according to first and second order moments of gradients. By penalizing high weights, weight decay reduces overfitting and encourages the development of broadly applicable models.

2) Lion

A novel optimization algorithm designed specifically for deep learning models is called Evolved Sign Momentum (LION) [25]. Unlike AdamW, which stores entire gradients, LION only stores the gradient's moving average, placing a higher priority on computational simplicity and memory efficiency. It uses the gradient’s ‘sign' to determine direction, which could result in quicker convergence and better accuracy. LION leverages two different momentum factors to improve update efficiency and momentum tracking. In a variety of image classification tasks, it has shown comparable or better performance than well-known optimizers like Adam and Adafactor [26], frequently with shorter training times and less computational load. LION offers an attractive substitute for deep learning optimization, and given its ease of use and encouraging outcomes, it is anticipated that its use in the field will grow.

IV. PROPOSED METHODOLOGY

Our approach involved utilizing pre-trained neural networks as feature extractors. Transfer learning, a common practice, entails employing a model previously trained on an extensive dataset for a particular task, such as image classification[27], natural language processing, or object detection[28]. These pre-trained models capture rich and generic features from their original training data, rendering them valuable for similar tasks. To employ a pre-trained model as a feature extractor, one typically removes the final layers, or a subset thereof, which are task-specific. The earlier layers are retained due to their acquisition of generic features applicable to various tasks. Once the final layers are removed, the weights of the remaining layers are frozen, implying that these weights remain unchanged during the training of the new model. The objective is to preserve the knowledge encoded by the pre-trained model and exclusively fine-tune the newly added layers. Additional layers tailored to the specific task are then incorporated into the model. For instance, in the case of utilizing a pre-trained image classification model, a few dense layers might be added for classification. The output of the pre-trained layers functions as the extracted features employed by the new layers for the defined task. Transfer learning as a feature extractor[29] offers the advantage of leveraging the knowledge acquired by a model on a large dataset, even if the dataset for the specific task is relatively small. This approach often leads to quicker convergence and superior performance compared to training a model entirely from scratch.

In our specific study, we removed the classification layers of the pre-trained model, substituting them with a dedicated classification head, as depicted in Figure [4]. This modification aimed to singularly focus on training the classifier and observe the impact of different models and optimizers on the convergence behaviour and overall accuracy of each test case.

By isolating the training of the classifier, we sought to analyse and compare the effects of various models and optimizers on the performance metrics. This approach allows for a more targeted investigation into how changes in the classification layers, along with the choice of optimizer and model architecture, influence the convergence dynamics and the ultimate accuracy across different test cases. The distinctive configuration illustrated in Figure [4] facilitates a clear understanding of the modifications made to the pre-trained model and their impact on the specific task at hand.

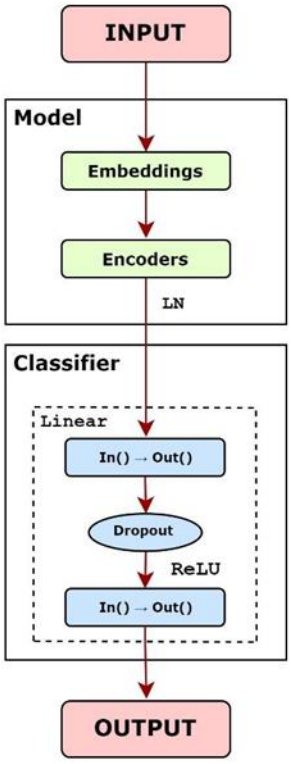

Figure 2: Diagrammatic Representation of Proposed Methodology

Table 3: Tabular Comparison between Model Architecture

Model	**ConvNeXtV2-base**	**ViT-base**
Trainable Parameters	272,166	206,630
Non-Trainable Parameters	87,692,800	85,798,656
Total Parameters	87,964,966	86,005,286
Year Released	2023	2021

V. Evaluation Metrics

Accuracy, precision, recall, and F1 score are among the metrics used to evaluate the performance of the model. The following formula is used to calculate accuracy, which is the proportion of correctly predicted images. The categorical cross-entropy loss is applied in scenarios where there are more than two classes. If C denotes the number of classes, and y_i is a one-shot encoded vector while y_i signifies the anticipated probability of class i, the loss is calculated as:

$$L(y, \hat{y}) = \sum_{i=1}^{C} y_i \log(y_i) \qquad \text{Eq. (1)}$$

Precision measures the ratio of correctly predicted positive results to the total positive results predicted by the model. Recall determines the number of correctly predicted positive instances by comparing true positive results to the total samples. The F1 score, another critical metric, evaluates model performance by computing the weighted harmonic mean of precision and recall.

VI. Results and Discussion

This section presents an analysis of the models' performance in our experimental setup across 20 epochs of training. We provide insights into convergence, flexibility, and optimization in particular scenarios. There are notable improvements in accuracy over epochs for both the ViT and ConvNeXtV2 models (Table 4, Figure 3). After Adam optimization, the ViT model's accuracy increases from 89.02% to 99.48%. The final accuracy is the same, but it starts lower (70.65%) with Lion optimization. While the Lion-optimized version begins at 68.67% and ends at 98.47%, the ConvNeXtV2 model, using Adam optimization, starts at 78.58% and ends at 99.02%.

Table 4: Tabular Comparison between Training Accuracy for Non-Augmented Dataset

Epoch	**ViT**		**ConvNeXtV2**	
	Adam	**Lion**	**Adam**	**Lion**
1	89.02%	70.65%	78.58%	68.67%
2	96.69%	94.15%	92.76%	92.05%
3	97.81%	97.22%	95.02%	95.53%
4	98.11%	98.12%	95.97%	96.66%
5	98.31%	98.57%	96.63%	97.29%
6	98.74%	98.80%	97.09%	97.56%
7	98.76%	98.92%	97.50%	97.86%
8	98.92%	99.13%	97.71%	97.92%
9	99.02%	99.18%	97.78%	98.10%
10	99.07%	99.19%	98.03%	98.23%
11	99.12%	99.23%	98.37%	98.34%

12	99.24%	99.34%	98.33%	98.29%
13	99.24%	99.34%	98.53%	98.32%
14	99.26%	99.35%	98.58%	98.23%
15	99.32%	99.41%	98.58%	98.36%
16	99.29%	99.40%	98.77%	98.47%
17	99.40%	99.47%	98.83%	98.50%
18	99.42%	99.44%	98.93%	98.46%
19	99.36%	99.48%	98.94%	98.46%
20	99.48%	99.48%	99.02%	98.47%

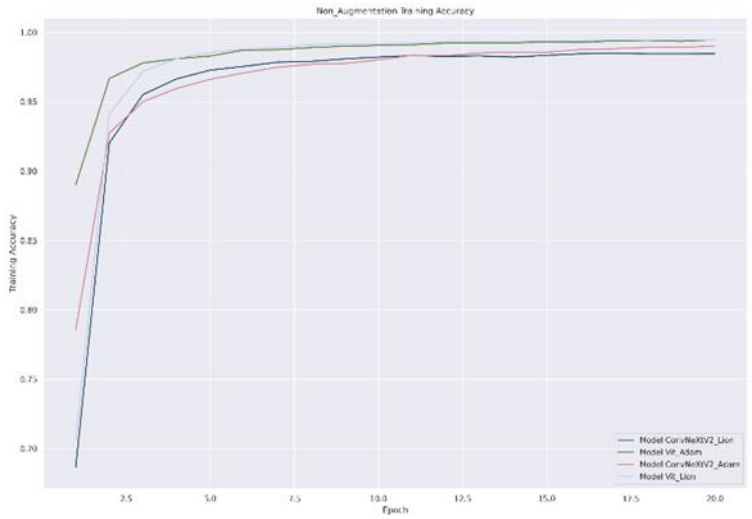

Figure 3: Graphical Representation of Training Accuracy on Non-Augmented Dataset

Table 5 and Figure 4 present the analysis of a consistent loss reduction over 20 epochs, suggesting continuous learning and improvement. We find differences between the models: ViT (Adam) and ConvNeXtV2_Adam shows lower initial losses and faster convergence than ViT (Lion) and ConvNeXtV2_Lion, probably because of differences in architecture or optimization. Every model exhibit stability and convergence, pointing to a dependable learning procedure. Even though ViT (Adam) and ConvNeXtV2_Adam shows marginally lower final loss values, a thorough analysis taking metrics like validation performance and accuracy into account is necessary. These results point to possible avenues for fine-tuning, especially for models with larger initial losses, by experimenting with different architectures or modifying hyper-parameters.

Table 5: Tabular Comparison between Training Loss for Non-Augmented Dataset

Epoch	**ViT**		**ConvNeXtV2**	
	Adam	**Lion**	**Adam**	**Lion**
1	0.4016	1.1941	0.8197	1.1985
2	0.1046	0.1766	0.2462	0.2385
3	0.0697	0.0843	0.1669	0.1328
4	0.0566	0.0555	0.1325	0.0964
5	0.0485	0.0438	0.1091	0.0776
6	0.0383	0.0348	0.0952	0.0694
7	0.0371	0.0322	0.0813	0.0610
8	0.0334	0.0256	0.0738	0.0588

9	0.0291	0.0244	0.0697	0.0552
10	0.0266	0.0237	0.0614	0.0518
11	0.0256	0.0224	0.0539	0.0478
12	0.0225	0.0188	0.0506	0.0485
13	0.0210	0.0173	0.0464	0.0466
14	0.0214	0.0192	0.0437	0.0496
15	0.0199	0.0169	0.0441	0.0464
16	0.0192	0.0180	0.0387	0.0433
17	0.0171	0.0158	0.0370	0.0426
18	0.0165	0.0156	0.0334	0.0425
19	0.0175	0.0152	0.0334	0.0440
20	0.0155	0.0156	0.0306	0.0424

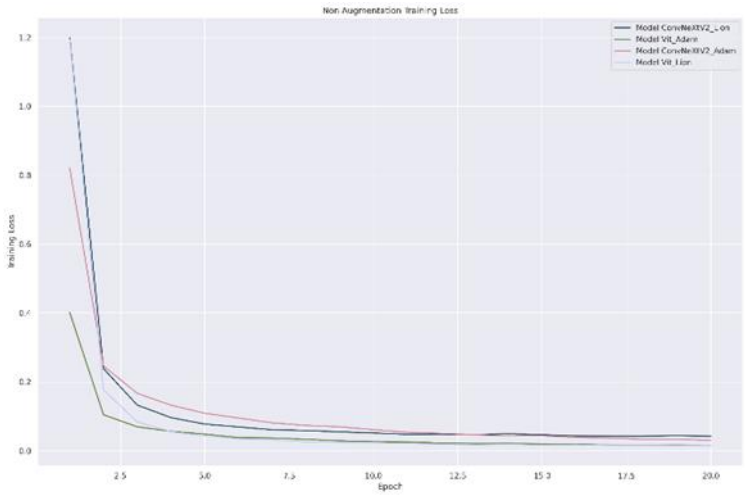

Figure 4: Graphical Representation of Training Loss on Non-Augmented Dataset

As shown in Table 6 and Figure 5, the analysis of the provided accuracy values for four models, namely ViT (Adam), ViT (Lion), ConvNeXtV2_Adam, and ConvNeXtV2_Lion, over 20 epochs reveal strong overall performance with consistent accuracy well above 98% by the 20th epoch. Stability is observed in the consistent increase of accuracy without significant fluctuations. ViT (Adam) and ConvNeXtV2_Adam starts with slightly higher accuracy but closely parallel the trends of ViT (Lion) and ConvNeXtV2_Lion, resulting in comparable final accuracy values. The analysis suggests that these models are reliable, demonstrating fine-tuning opportunities to optimize performance based on specific requirements or constraints.

Table 6: Tabular Comparison between Training Accuracy for Augmented Dataset

Epoch	**ViT**		**ConvNeXtV2**	
	Adam	**Lion**	**Adam**	**Lion**
1	88.28%	71.71%	79.05%	69.99.%
2	96.21%	94.42%	92.71%	92.61%
3	97.27%	96.99%	94.55%	95.58%
4	97.74%	97.83%	95.79%	96.42%
5	98.05%	98.30%	96.33%	96.95%
6	98.32%	98.52%	96.83%	97.36%

7	98.43%	98.77%	97.31%	97.50%
8	98.61%	98.78%	97.41%	97.65%
9	98.73%	98.90%	97.66%	97.80%
10	98.79%	98.91%	97.80%	97.91%
11	98.82%	99.07%	97.96%	97.95%
12	98.98%	99.12%	98.19%	98.01%
13	99.08%	99.11%	98.30%	98.08%
14	99.05%	99.15%	98.38%	98.05%
15	99.07%	99.24%	98.51%	98.14%
16	99.16%	99.22%	98.50%	98.11%
17	99.15%	99.26%	98.67%	98.16%
18	99.19%	99.16%	98.75%	98.25%
19	99.24%	99.24%	98.80%	98.24%
20	99.26%	99.35%	98.83%	98.24%

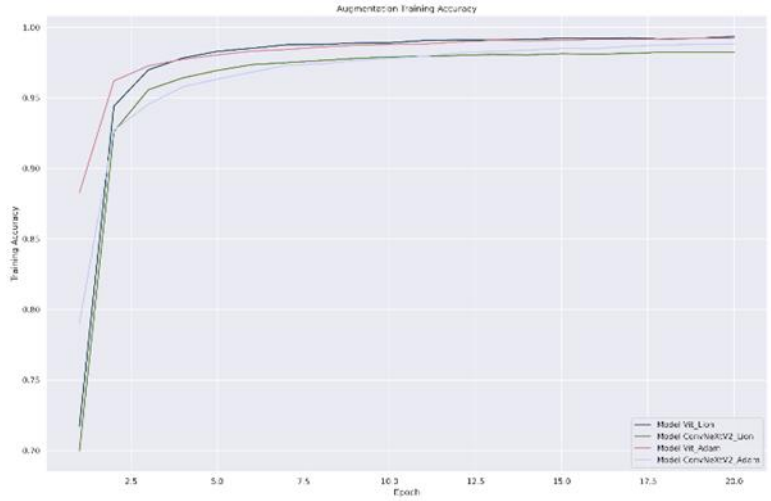

Figure 5: Graphical Representation of Training Accuracy on Augmented Dataset

In Table 7 and Figure 6, the analysis of the provided loss values for four models—ViT (Adam), ViT (Lion), ConvNeXtV2_Adam, and ConvNeXtV2_Lion—over 20 epochs reveal a consistent decrease in loss, indicating effective learning and improved performance. The decreasing trend across all models signifies continuous improvement, with the models adjusting their parameters to better fit the training data. Although ViT (Adam) and ConvNeXtV2 (Adam) start with slightly lower initial loss values, all models exhibit a similar rate of improvement and converge towards optimal states, reflecting stable and reliable learning processes. The final loss values are close among the models, suggesting comparable performance.

Table 7: Tabular Comparison between Training Loss for Augmented Dataset

Epoch	**ViT**		**ConvNeXtV2**	
	Adam	**Lion**	**Adam**	**Lion**
1	0.4133	1.0990	0.7877	1.1164
2	0.1167	0.1667	0.2436	0.2217

3	0.0809	0.0883	0.1743	0.1293
4	0.0685	0.0637	0.1350	0.1011
5	0.0570	0.0510	0.1150	0.0861
6	0.0486	0.0421	0.1002	0.0747
7	0.0452	0.0375	0.0858	0.0716
8	0.0401	0.0346	0.0800	0.0663
9	0.0363	0.0305	0.0729	0.0634
10	0.0338	0.0303	0.0682	0.0589
11	0.0332	0.0260	0.0607	0.0577
12	0.0280	0.0255	0.0559	0.0554
13	0.0263	0.0245	0.0515	0.0548
14	0.0271	0.0235	0.0495	0.0540
15	0.0247	0.0210	0.0457	0.0519
16	0.0243	0.0220	0.0447	0.0527
17	0.0254	0.0205	0.0396	0.0529
18	0.0231	0.0228	0.0374	0.0508
19	0.0221	0.0204	0.0361	0.0504
20	0.0205	0.0184	0.0345	0.0498

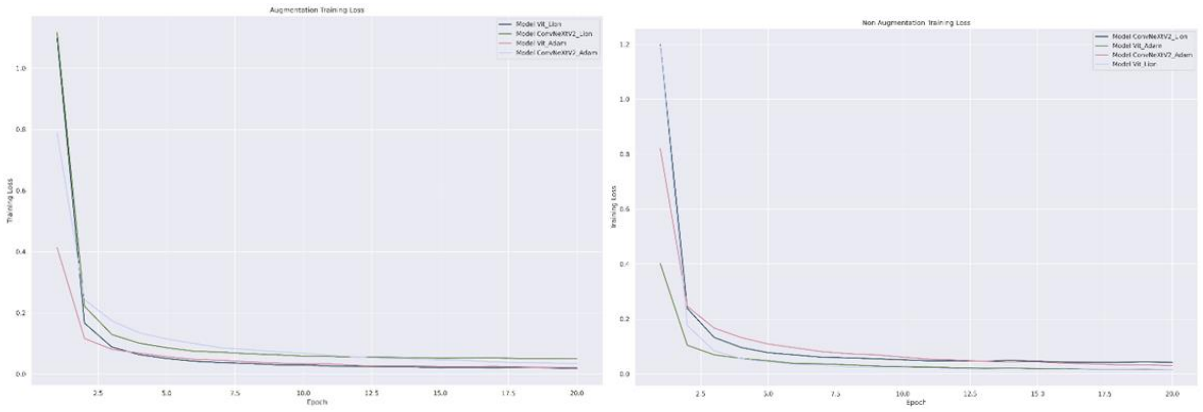

Figure 6: Graphical Representation of Training Loss on Augmented Dataset

A thorough analysis (Table 8) that included a range of test cases and measured the model's performance on the validation set and its ability to generalize to new data was carried out using metrics like accuracy, loss, precision, recall, and F1-score. The efficacy of AdamW is suggested by the slightly lower loss values found in the analysis of optimizer selection, especially when comparing it to Lion. Though fine-tuning might be required for optimal performance, both learning rates (0.001 for Lion and 0.0001 for AdamW) contribute to good convergence. The ViT models outperform the ConvNeXtV2 model in terms of loss, accuracy, and training time, highlighting the influence of optimizer and model selection on resource consumption. In this particular context, ViT models, especially with AdamW, show promise. Overall, the choice of optimizer and learning rate exerts a significant influence on model performance.

Table 8: Evaluation Metrics of Various Test Cases for Non-Augmented Dataset

Model	**Optimizer**	**Learning Rate**	**Loss**	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Training Time**
ConvNext V2	AdamW	0.0001	0.0545	0.9822	0.9822	0.9822	0.9821	218m 14s
	Lion	0.001	0.0519	0.9836	0.9839	0.9836	0.9836	217m 32s
ViT	AdamW	0.0001	0.0432	0.9884	0.9888	0.9884	0.9885	148m 42s
	Lion	0.001	0.0389	0.9882	0.9883	0.9882	0.9882	149m 3s

Table 9: Evaluation Metrics of Various Test Cases for Augmented Dataset

Model	Optimizer	Learning Rate	Loss	Accuracy	Precision	Recall	F1 Score	Training Time
ConvNext V2	AdamW	0.0003	0.0611	0.9795	0.9796	0.9795	0.9795	257m 43s
	Lion	0.001	0.0526	0.9832	0.9834	0.9832	0.9831	260m 5.3s
ViT	AdamW	0.0003	0.0553	0.9839	0.9843	0.9839	0.984	180m 47s
	Lion	0.001	0.0541	0.9834	0.9835	0.9834	0.9833	179m 19s

The results presented in Table 9 consistently highlight the efficacy of the AdamW optimizer. Specifically, models that use it—ConvNeXtV2 with AdamW and ViT with AdamW—achieve marginally lower loss values than models that use the Lion optimizer. Reasonable convergence is seen despite different learning rates (0.0003 for AdamW and 0.001 for Lion), though fine-tuning might be necessary for different tasks. High accuracy and balanced precision, recall, and F1 scores are displayed by all models, demonstrating their resilience to accurate predictions. ViT models—both the Lion and AdamW variants—perform better than ConvNeXtV2 models in a number of metrics, indicating either their applicability for the task at hand or the necessity of additional ConvNeXtV2 optimization. ViT models are particularly effective in resource-constrained settings, as evidenced by their shorter training times. While potential fine-tuning avenues—such as experimenting with different learning rates, investigating alternative optimizers, or adjusting hyperparameters—could further boost performance or shorten training duration, consistency in metrics across models reinforces model stability and generalization

In summary, the observations regarding optimizer and learning rate impact remain consistent, with ViT models showing slightly better performance and shorter training times compared to ConvNeXtV2 models, prompting further exploration of fine-tuning opportunities based on specific requirements.

VII. Conclusion

Through a thorough examination of model performance, optimizer selection, and learning rates across different epochs and metrics, we gain valuable insights into the effectiveness of the Lion optimizer in the context of ViT and ConvNeXtV2 models. According to the LION's Proposed paper[25], Lion faces challenges in optimizing the ConvNet architecture, ConvNeXtV2, but excels in optimizing transformer-based architecture, ViT, when compared to AdamW. Despite achieving only a slight improvement in accuracy, Lion demonstrates the ability to converge faster than AdamW. This underscores the robustness of the Lion optimizer in facilitating efficient learning and refinement of representations during training. The consistent rise in accuracy, coupled with stable convergence, suggests that the Lion optimizer contributes to a reliable and consistent learning process for ViT models. In contrast, ConvNeXtV2 models, while showing effective learning with both Adam and Lion optimizers, follow slightly different trajectories in terms of starting accuracy and convergence rates. The Lion-optimized ConvNeXtV2 model, starting with lower accuracy, manages to reach a final accuracy close to its Adam-optimized counterpart, highlighting the adaptability and effectiveness of the Lion optimizer in enhancing ConvNeXtV2 model performance. The consistent convergence and competitive final accuracy values position the Lion optimizer as a viable choice for optimizing these models. While further exploration and fine-tuning opportunities may be considered based on specific requirements, the results affirm the efficacy of the Lion optimizer in contributing to the success of deep learning models in this particular context.

VIII. References

[1] Y. Zhao et al., "Plant Disease Detection Using Generated Leaves Based on DoubleGAN," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 19, no. 3, pp. 1817-1826, 1 May-June 2022, doi: 10.1109/TCBB.2021.3056683.

[2] Too, Edna & Li, Yujian & Njuki, Sam & Yingchun, Liu. (2018). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture. 161. 10.1016/j.compag.2018.03.032.

[3] Hassan, Sk Mahmudul, Arnab Kumar Maji, Michał Jasiński, Zbigniew Leonowicz, and Elżbieta Jasińska. 2021. "Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach" Electronics 10, no. 12: 1388. https://doi.org/10.3390/electronics10121388.

[4] B. Chellapandi, M. Vijayalakshmi and S. Chopra, "Comparison of Pre-Trained Models Using Transfer Learning for Detecting Plant Disease," 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Greater Noida, India, 2021, pp. 383-387, doi: 10.1109/ICCCIS51004.2021.9397098.

[5] Kaya, A, Keceli, AS, Catal, C, Yalic, HY, Temucin, H & Tekinerdogan, B 2019, 'Analysis of transfer learning for deep neural network-based plant classification models', Computers and Electronics in Agriculture, vol. 158, pp. 20-29. https://doi.org/10.1016/j.compag.2019.01.041.

[6] M. Ahmad, M. Abdullah, H. Moon and D. Han, "Plant Disease Detection in Imbalanced Datasets Using Efficient Convolutional Neural Networks With Stepwise Transfer Learning," in IEEE Access, vol. 9, pp. 140565-140580, 2021, doi: 10.1109/ACCESS.2021.3119655.

[7] S. Pudumalar, S. Muthuramalingam, Hydra: An ensemble deep learning recognition model for plant diseases, Journal of Engineering Research, 2023, ISSN 2307-1877, https://doi.org/10.1016/j.jer.2023.09.033.

[8] Borhani Y, Khoramdel J, Najafi E. A deep learning-based approach for automated plant disease classification using vision transformer. Sci Rep. 2022 Jul 7;12(1):11554. doi: 10.1038/s41598-022-15163-0. PMID: 35798775; PMCID: PMC9262884.

[9] R. S. Sandhya Devi1, V. R. Vijay Kumar, P. Sivakumar. (2022). EfficientNetV2 Model for Plant Disease Classification and Pest Recognition. Computer Systems Science and Engineering 2023, 45(2), 2249-2263. https://doi.org/10.32604/csse.2023.032231.

[10] Hughes, D.P. and Salathe, M. (2015) An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics. arXiv:1511.08060. http://arxiv.org/abs/1511.08060.

[11] Li M, Li CG, Guo J. Cluster-Guided Asymmetric Contrastive Learning for Unsupervised Person Re-Identification. IEEE Trans Image Process. 2022;31:3606-3617. doi: 10.1109/TIP.2022.3173163. Epub 2022 May 26. PMID: 35576408.

[12] Rico & Fryzlewicz, Piotr, (2016). "Random rotation ensembles," LSE Research Online Documents on Economics 62182, London School of Economics and Political Science, LSE Library.

[13] Khalifa, Nour Eldeen & Loey, Mohamed & Mirjalili, Seyedali. (2022). A comprehensive survey of recent trends in deep learning for digital images augmentation. Artificial Intelligence Review. 55. 10.1007/s10462-021-10066-4.

[14] Liu, Z, Mao, H, Wu, CY, Feichtenhofer, C, Darrell, T & Xie, S 2022, A ConvNet for the 2020s. in Proceedings - 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2022-June, IEEE Computer Society, pp. 11966-11976, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, United States, 6/19/22. https://doi.org/10.1109/CVPR52688.2022.01167.

[15] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, Saining Xie; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, pp. 16133-16142.

[16] Touvron, Hugo & Cord, Matthieu & Sablayrolles, Alexandre & Synnaeve, Gabriel & Jégou, Hervé. (2021). Going deeper with Image Transformers. 32-42. 10.1109/ICCV48922.2021.00010.

[17] Dan Hendrycks, Kevin Gimpel. (2023). Gaussian Error Linear Units (GELUs). https://doi.org/10.48550/arXiv.1606.08415v5.

[18] Abien Fred Agarap. (2019). Deep Learning using Rectified Linear Units (ReLU). https://doi.org/10.48550/arXiv.1803.08375v2.

[19] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. https://doi.org/10.48550/arXiv.2010.11929.

[21] Rasche, Christoph. (2019). Computer Vision. Bucharest: Polytechnic University of Bucharest.

[22] Khurana, Diksha & Koli, Aditya & Khatter, Kiran & Singh, Sukhdev. (2022). Natural Language Processing: State of The Art, Current Trends and Challenges. Multimedia Tools and Applications. 82. 10.1007/s11042-022-13428-4.

[23] Ilya Loshchilov, Frank Hutter. (2019). Decoupled Weight Decay Regularization. https://doi.org/10.48550/arXiv.1711.05101v3.

[24] Gnecco, Giorgio & Sanguineti, Marcello. (2009). The weight-decay technique in learning from data: An optimization point of view. Computational Management Science. 6. 53-79. 10.1007/s10287-008-0072-5.

[25] Chen, Xiangning, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu and Quoc V. Le. “Symbolic Discovery of Optimization Algorithms.” ArXiv abs/2302.06675 (2023): n. pag.

[26] Noam Shazeer, Mitchell Stern. (2018). Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. https://doi.org/10.48550/arXiv.1804.04235.

[27] Krishna, M & Neelima, M & Mane, Harshali & Matcha, Venu. (2018). Image classification using Deep learning. International Journal of Engineering & Technology. 7. 614. 10.14419/ijet.v7i2.7.10892.

[28] Vaishnavi, K. & Reddy, G. & Reddy, T. & Iyengar, N. & Shaik, Subhani. (2023). Real-time Object Detection Using Deep Learning. Journal of Advances in Mathematics and Computer Science. 38. 24-32. 10.9734/jamcs/2023/v38i81787.

[29] Jha, Ritesh & Bhattacharjee, Vandana & Mustafi, Abhijit. (2022). Transfer Learning with Feature Extraction Modules for Improved Classifier Performance on Medical Image Data. Scientific Programming. 2022. 1-10. 10.1155/2022/4983174.

[30] Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: theoretical analysis and applications. In Proceedings of the 40th International Conference on Machine Learning (ICML'23), Vol. 202. JMLR.org, Article 992, 23803–23828.

[31] Andreieva, Valeria & Shvai, Nadiia. (2021). Generalization of Cross-Entropy Loss Function for Image Classification. Mohyla Mathematical Journal. 3. 3-10. 10.18523/2617-7080320203-10.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. (2023). Attention Is All You Need. https://doi.org/10.48550/arXiv.1706.03762v7.