

# Enhanced Plant Disease Classification through Transfer Learning with Modified VGGNet and Inception Modules

Bashir Abubakar, Abdulfatai Dare Adekale, Risikat Folashade Adebisi, Sani Isah Abubakar, Rufai Hassan, and Habeeb Bello-Salau

**Abstract**—Advances in technology and high computational power have driven extensive use of Machine Learning (ML) and Deep Learning (DL) models in diverse fields, such as object recognition, intelligent transportation, and precision agriculture, with particular focus on plant disease classification. Literature highlights that DL models, initially designed for large datasets like ImageNet, often lack efficiency for tasks with smaller datasets and fewer classes, necessitating architectural modifications for better performance. This study addresses this issue by exploring optimized activation functions to overcome the vanishing gradient problem, which can impede model training and accuracy in small-scale datasets. Specifically, this research modifies VGG-16 and VGG-19 architectures, replacing the fifth convolutional layer with an additional 3x3x512 convolutional layer and integrating the ELU activation function instead of Swish. Two inception layer modules and a global pooling layer were also added, followed by softmax for classification. Trained on curated maize and rice datasets, the proposed model achieved 98.40% accuracy, surpassing the Swish function's 97.97%. This improvement underscores the ELU function's effectiveness in improving model performance, with practical applications in early plant disease detection, aiding precision agriculture by enabling timely intervention and yield protection.

**Index Terms**—Activation functions, Exponential Linear Unit, Disease, Machine Learning, Plant, Transfer Learning .

## I. INTRODUCTION

Rice and maize are the most widely cultivated grains globally, driven by high demand for consumption and industrial processing [1], [2]. However, these crops are highly susceptible to diseases, reducing farm output [3]. Early disease detection and treatment can significantly enhance farm productivity and boost farmers income. For local farmers, manual disease classification is often challenging, and expert

advice may be inaccessible [4]. This highlights the need for automated disease classification using artificial intelligence (AI) methods [5], [6]. AI has become essential in precision agriculture, particularly in deploying deep learning (DL) models for plant disease classification [7]. Early detection through these models can minimize the negative impact of diseases on yield.

Initially, classical machine learning methods, such as SVM, KNN, and Random Forests, were used for plant disease detection [8], [9]. However, recent advances in Convolutional Neural Networks (CNNs) have expanded the use of pretrained DL models [10]. The performance of DL models in disease classification depends on factors like activation functions, model architecture, and dataset quality [11]. For instance, VGGNet, a popular architecture, is dense with neurons, which can slow training and affect classification performance. Studies show that modifying model layers, particularly the top layers can improve efficiency, reducing neurons and training time [4]. While ReLU is the traditional activation function, alternatives like LeakyReLU, PReLU, SELU, and ELU have shown better performance by mitigating the “dying neuron” problem [12]. Although some studies have optimized VGGNet for plant disease classification [7], [4], they have not extensively explored the impact of different activation functions on model performance. For example, recent work reported in [4] replaced VGGNet’s last layers with a 3x3x512 convolutional layer, batch normalization, and substituted ReLU with Swish. However, other activation functions were not examined.

Building on this foundation, our study modifies VGGNet as per [4] and evaluates the performance of activation functions that address the vanishing gradient problem, including ReLU, ELU, SELU, Swish, and LeakyReLU. The rest of the paper is structured as follows: Section 2 review related work on DL models with emphasis of their activation functions, Section 3

Manuscript received November 10, 2024. This work was supported by Tertiary Education Trust Fund (TETFUND), Nigeria, for funding this research as part of the 2021 National Research Fund (NRF) grant cycle, under the project titled "A Novel Artificial Intelligence of Things (AIoT) Based Assistive Smart Glass Tracking System for the Visually Impaired" (TETF/DR&D/CE/NRF2021/SETI/ICT/00140/VOL.1).

B. Abubakar is with the Computer Engineering Department, Federal Polytechnic Bali, Taraba state, Nigeria (e-mail: abubakarbashir28@yahoo.com).

A. D. Adekale is with the Computer Engineering Department, Ahmadu Bello University, Zaria-Nigeria (e-mail: adekale@abu.edu.ng)

R. F. Adebisi is with the Computer Engineering Department, Ahmadu Bello University, Zaria-Nigeria (e-mail: rfadebiyi@abu.edu.ng)

S. I. Abubakar is with the Agricultural and Bio-Environmental Engineering Department, Federal Polytechnic Bali, Taraba state, Nigeria (e-mail: sanisaka@yahoo.com).

R. Hassan is with the Computer Engineering Department, Federal Polytechnic Bali, Taraba state, Nigeria (e-mail: ruphae Hassan@gmail.com).

H. Bello-Salau is with the Computer Engineering Department, Ahmadu Bello University, Zaria-Nigeria (phone: +2348036816031; e-mail: bellosalau@abu.edu.ng).



presents the proposed methodology, while section 4 present the results and discussion and section 5 concludes the paper.

## II. RELATED WORK

### A. Activation functions and vanishing gradient

In training deep learning networks, particularly those with many layers, the vanishing gradient problem, where gradients become very small often arises, slowing learning as layer weights update slowly during backpropagation [13], [14]. This hinders the network's ability to learn meaningful features from the data. To counter this, various activation functions have been proposed, with the Rectified Linear Unit (ReLU) being especially effective. However, newer functions such as LeakyReLU, Swish, Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SELU), and Parametric ReLU (PReLU) further address issues like the "dying neurons" problem, potentially outperforming ReLU in deep networks.

The Rectified Linear Unit (ReLU), is a non-linear activation function commonly used in deep learning due to its computational efficiency and ability to handle complex data [15]. By mapping negative values to zero, it reduces complexity and helps prevent the vanishing gradient problem, improving network performance. However, ReLU has a "dying neuron" issue, where negative neurons remain inactive, potentially affecting model performance. The function is mathematically expressed in equation (1)

$$f(x) = \max(0, x) \quad (1)$$

The Leaky Rectified Linear Unit (LeakyReLU), is a modified version of ReLU that addresses the "dying neuron" issue by allowing a small positive gradient for negative inputs, keeping neurons active throughout training [16]. This small gradient for negative values helps maintain information flow but adds slight computational cost compared to ReLU. LeakyReLU is defined as presented in equation (2)

$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha \cdot x & \text{if } x < 0 \end{cases} \quad (2)$$

Also, the swish function, is an activation function similar to LeakyReLU, allowing small positive values for negative inputs but utilizing a sigmoid function [15]. It has an activation range from -3 to +3, enabling a broad input range. Swish often outperforms ReLU in deep networks, although its effectiveness can vary with model architecture and dataset characteristics. While more computationally intensive than ReLU, Swish is simple to implement as it does not add learning parameters. The Swish function is shown in equation (3)

$$\text{swish}(x) = x \cdot \text{sigmoid}(x) \quad (3)$$

Similarly, the Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SELU), and Parametric Rectified Linear Unit (PReLU) are activation functions designed to address issues in ReLU, such as the "dying neurons" problem []. ELU uses a negative saturation value controlled by a hyperparameter to maintain non-zero gradients for negative inputs, aiding convergence [17]. It is expressed mathematically as shown in equation (4)

$$\text{ELU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha \cdot (\exp(x) - 1) & \text{if } x < 0 \end{cases} \quad (4)$$

SELU builds on ELU by adding a scaling parameter, which regulates activation magnitude to prevent vanishing or exploding gradients [18]. It is expressed mathematically as presented in equation (5)

$$\text{SELU}(x) = \lambda \cdot \begin{cases} x & \text{if } x \geq 0 \\ \alpha \cdot (\exp(x) - 1) & \text{if } x < 0 \end{cases} \quad (5)$$

PReLU introduces a learnable parameter,  $\alpha_b$ , for the negative slope, enhancing flexibility in mapping data correlations but increasing the risk of overfitting if not well-regularized [8]. The PReLU equation is expressed in (6):

$$\text{PReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha_b \cdot x & \text{if } x < 0 \end{cases} \quad (6)$$

### B. Related work on plant Diseases Classification

Several studies have investigated how the architectures and functions relates to the performance of DL models. For example, the authors in [19] fine-tuned the VGGNET Convolutional neural network having sigmoid function as its classifier. The model had eight convolutional neural network layers and two fully connected layers. The convolutional layers have lesser feature maps when compared to the original VGGNET. Their model had an accuracy of 95.5% and was almost nine times faster when compared to the original VGGNET architecture. Although, the model was an end-to-end model, but has not incorporated inception module and used two fully connected layers which usually, are dense layers. Also, no investigation was done on the effects of other activation functions on the VGG model apart from the usual ReLU function. Numerous articles used different architectures approach for plant disease classification, but do not consider the effect of activation functions on the performance of their DL models.

There are numerous other articles which investigates the performance of DL models on plants using different architectures and other popular activation functions. Most of the articles which considered activation functions only compared with ReLU functions. The authors in [20] proposed cucumber disease detection and classification based on deep features selection and improved saliency method. Their model was achieved using five stages including: image enhancement, infected spots segmentation, deep features extraction, feature selection, and lastly classification. The images were segmented with model's novel "Sharif saliency-based (SHSB)" method. The segmented images were fed to deep learning's VGG16 and VGG19 for feature extraction; these features were then passed to multi-class support vector machine for disease classification. The model was compared with three models. The author's model's result was superior to the three compared models with accuracy above 96% and recognition rate of 98.4% and an execution time of 16.5seconds. However, the models used the dense fully connected, although the performance of different traditional machine learning classifiers were investigated but it does not look into an angle of the effects of activations functions on performance. The authors in [4] proposed an approach that involve the substitution of VGGNET last



convolutional layers with extended convolution Layer of  $3 \times 3 \times 512$ , Batch Normalization (BN) was added and ReLU function was also substituted with swish function. The authors also added two Inception module layers and a maxpooling layer substituted the fully connected layers. Model was based on rice disease, maize datasets, where rice dataset had higher result accuracy, with average of 92.0%. Although, this is an end-to-end model, there will be need to investigate the performance of other functions like LeakyRelu, ELU, PReLU used in the system.

Since many DL models, especially the VGGNet was modified for plant disease classification, there is a need to investigate the performance of different activation functions on the VGGNet modified architecture. Our study adopts [4] architecture to investigate different activation functions that addressed the vanishing gradient problem. These functions include the ReLU, ELU, SELU, Swish, and LeakyReLU.

### III. PROPOSED METHODOLOGY

This section is divided into two parts: the first details the dataset used, the second introduces the modified VGGNet model for plant disease classification, along with the performance evaluation metrics applied in the study.

#### A. Dataset Description

This research used two datasets. The first, downloaded from the publicly available Plant Village dataset (URL here), includes maize plant disease classifications. However, crop protection experts from the Institute of Agricultural Research (IAR) in Zaria, Nigeria, noted that some diseases in this dataset were misclassified, prompting the need for a curated dataset. Leveraging IAR experts' experience, a new dataset was curated for maize and rice diseases. Images were captured using a Canon 100 digital camera on experimental farms at the Division of Agricultural Colleges (DAC) of Ahmadu Bello University, Zaria, Nigeria. Four maize disease classes (northern leaf blight, bacterial leaf streak, common rust, and healthy) and three rice classes (leaf spot, leaf blight, and healthy) were documented, with each class containing at least 100 images. These images were pre-processed, entailing background removed, sharpened, and resized to 224x224 and augmented through techniques like rotation, flipping, shearing, and zooming. After augmentation, each class had 500 images, which were then used to train the modified VGG16 model.

#### B. Proposed Modified VGG-16 Model

This research examined various activation functions as reviewed in section IIA, within a modified VGGNet-16 model for plant disease classification. In the proposed approach, the fourth convolutional layer (conv5\_1) was replaced with a 3x3x512 convolution layer utilizing different activation functions to address vanishing gradient issues, and batch normalization was added to stabilize weight distribution across the network. Two inception modules were also incorporated, and the dense fully connected layers (FCL 1, 2, and 3) were replaced by a global pooling layer, followed by a softmax layer.

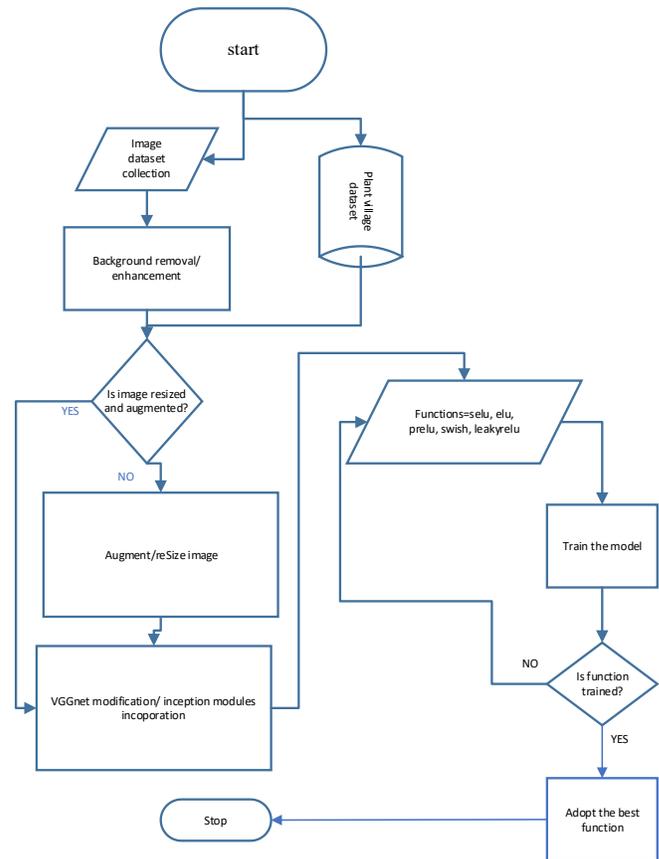


Figure 1: flowchart of the proposed Modified VGG-16 model.

This transfer learning-based model was applied to classify diseases in maize and rice plants, chosen for their high demand and importance in improving agricultural yields. The proposed methodology is summarized in figure 1

The performance of the proposed deep learning models is assessed using metrics such as accuracy, precision, recall, and F1-score, which are calculated based on the relationships between True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP refers to correctly classified positive samples, TN to correctly classified negative samples, FP to incorrectly classified positive samples, and FN to incorrectly classified negative samples. Precision (Pr) is the ratio of TP to the total predicted positive samples (TP + FP) as shown in Equation (7), while recall (Rc) is the ratio of TP to the total actual positive samples (TP + FN) as shown in Equation (8). Accuracy (A) is the ratio of correctly classified samples to the total samples (TP + TN) as shown in Equation (9). The F1-score (F1) is the harmonic mean of precision and recall, calculated using Equation (10).

$$P = \frac{TP}{TP+FP} \quad (7)$$

$$R = \frac{TP}{TP+FN} \quad (8)$$

$$A = \frac{TP+TN}{TP+FP+TN+FN} \quad (9)$$

$$F1 = 2 \times \frac{\text{top } Pr \times Rc}{Pr+Rc} \quad (10)$$

### IV. RESULTS AND DISCUSSION

This section presents the results from applying the proposed



model to both the curated maize and rice disease datasets and the Kaggle dataset. The evaluation includes the performance of various activation functions, specifically the ELU, SELU, PReLU, LeakyReLU, and Swish, all aimed at addressing vanishing gradient issues that can affect model accuracy and efficiency during training.

#### A. Result for Maize Diseases Classification

The results are organized into two sections: the first presents findings from the curated maize dataset, while the second outlines result from publicly available Kaggle datasets.

##### 1) Results obtained from the curated maize dataset

The dataset includes four classes: Blight, Spot, Common Rust, and Healthy. Due to space constraints, only the best performing activation function when incorporated into VGG-16 model was compared to similar work reported in [4], and reported in Table 1. Of the five activation functions tested, ELU achieved the best results with a training accuracy of 98.50% and validation accuracy of 97.50%. Figures 2a and 2b display the accuracy and loss graphs, respectively, showing a steady improvement in training accuracy (Figure 2a), indicating effective model learning likely due to the ELU function's performance in disease classification.

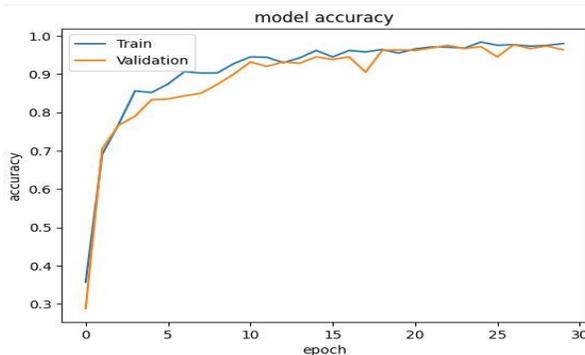


Figure 2a: Training and Validation Accuracy Graph for ELU

The second-highest performing activation function with the modified VGG-16 model was PReLU, achieving training and validation accuracies of 98.40% and 95.00%, respectively. As shown in Figure 3a, PReLU demonstrated strong early convergence, second only to ELU. This performance likely results from the model's ability to optimize the function's hyperparameter,  $\alpha$ , for the architecture and dataset early in training.

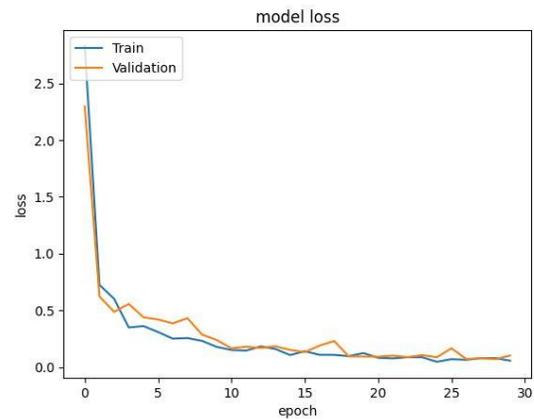


Figure 2b: Loss Graph for ELU

Training and validation losses were also low at 0.0438 and 0.1586 (Figure 3b), outperforming prior results of 0.0856 and 0.2409, respectively, as reported in [4].

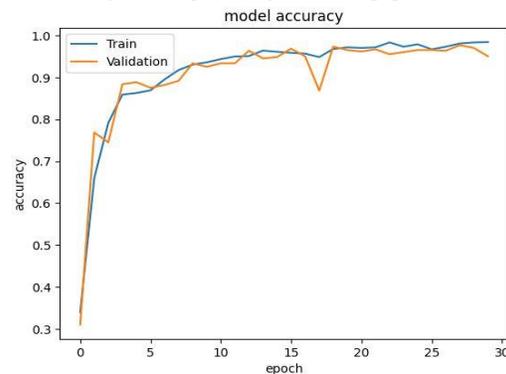


Figure 3a: Training and Validation Accuracy Graph for PReLU

The Swish function, following Chen et al. (2020), ranked third in performance with training and validation accuracies of 97.97% and 96.33%, respectively, as shown in Figure 5a. The model demonstrated

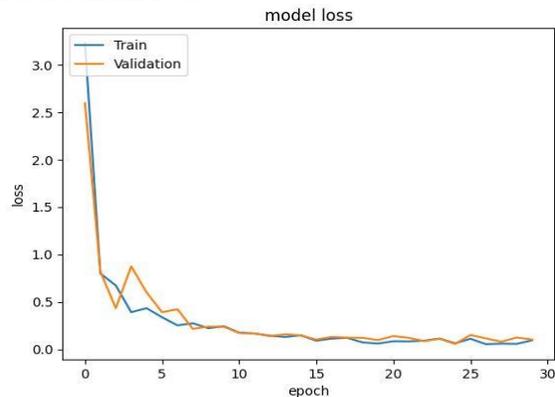


Figure 3b: Loss Graph for PReLU

early convergence around 15-18 epochs. Training and validation losses were 0.0561 and 0.0833 (Figure 5b), indicating strong predictive accuracy against expected outputs.

The SELU function's performance at early epochs was not stable (Figure 6a) as the gradient from the graph can be seen to be haphazard but steadily improves after 10 epochs. The sign for convergence of the model started after epoch 20. This shows



that the model had difficulties in adjusting its two hyperparameters, the  $\lambda$  and  $\alpha$  which were predetermined before the training. The training and validation accuracy of the model was 97.48% and 95.17% respectively. The training and validation loss of the function were 0.0670 and 0.8933 respectively (Figure 6b) which was more than the ELU, PReLU and Swish functions, this shows that the model's prediction against expected output was poor compared to ELU, PReLU and Swish functions.

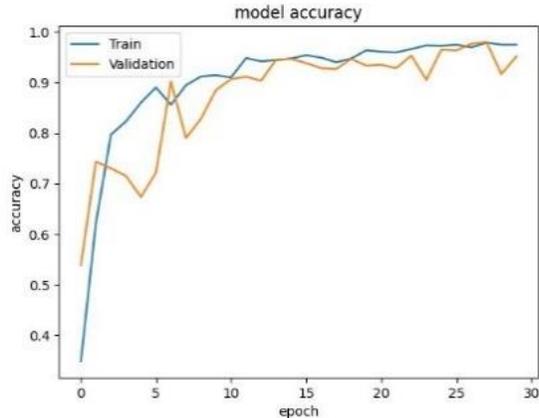


Figure 6a: Training and Validation Accuracy for Selu

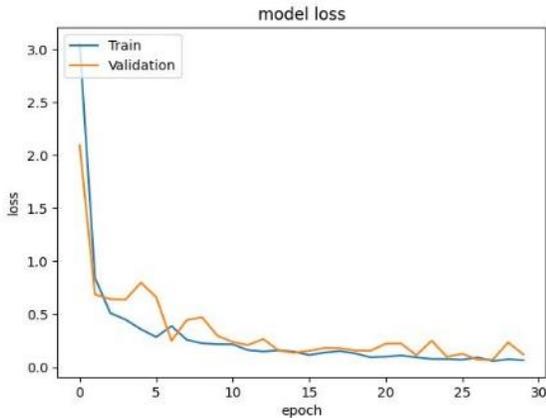


Figure 6b: Loss Graph for Selu

The LeakyRelu training and validation accuracies were 96.71% and 93.50% respectively. The function has shown steady

improvement in performance but could not converge early not until around 17-20 epochs (see Figure 7a). The training and validation loss of the LeakyRelu were 0.0786 and 0.1742 respectively which is lower than all the other four functions in comparison.

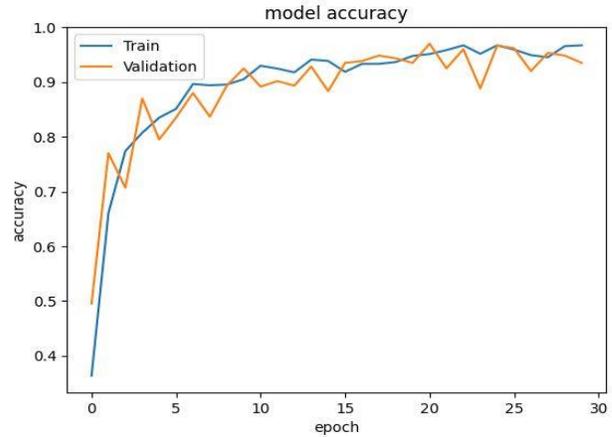


Figure 4.5a: Training and Validation Accuracy for LeakyRelu

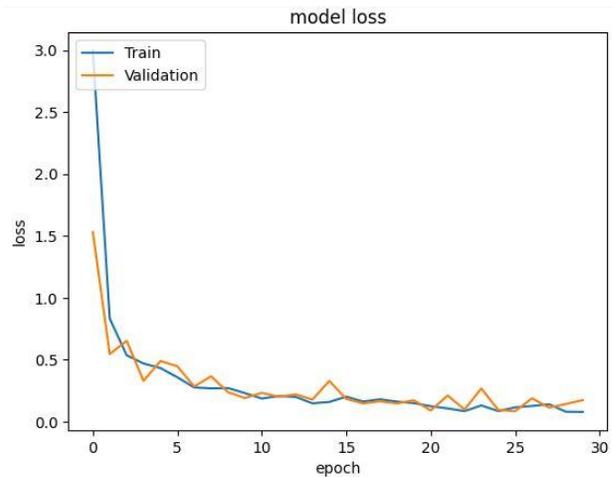


Figure 7b: Loss Graph for LeakyRelu

In summary, the performance of the model was influenced by factors such as the architecture of the model, the activation functions employed, and the nature of datasets used for training.

Table 1: Summary of Different Functions Performances on the Curated Maize Dataset

S/N	Function	Epoch 10				Epoch 30				Testing accuracy
		Training accuracy	Validation accuracy	Training Loss	Validation Loss	Training accuracy	Validation accuracy	Training Loss	Validation Loss	
1.	Elu	93.74	90.00	0.1775	0.2379	98.50	97.50	0.0446	0.1016	89.83
2.	Leaky-ReLU	90.51	92.51	0.2304	0.1906	96.71	93.50	0.0786	0.1742	90.83
3.	PReLU	93.56	92.50	0.1613	0.1764	98.40	95.00	0.0438	0.1586	92.30
4.	Swish	93.22	93.17	0.1717	0.1670	97.97	96.33	0.0561	0.833	91.17
5.	Selu	91.43	88.50	0.2177	0.2983	97.48	95.17	0.0670	0.1205	0.8933

The result of the best performing function which invariably is the ELU function, was compared with the INCORPORATED VGGNet. At epoch 10, the training and validation accuracy of both INC-VGGNet and the ELU functions of the developed

model of this study were nearly the same (Figure 8a). At epoch 30, the developed model of this study considerably outperformed the INC-VGGNet in



both the training and validation accuracies. This can be attributed to the ELU's function ability to adapt to the problem of dying neurons, better than the swish functions in this particular architecture and dataset during the training. The ELU function showed its superiority over the INC-VGGNet even better when the training and validation loss is put into consideration as shown in Figure 8b where the margin is clearly seen. This shows the model was able to represent the classification problem at hand better compared to the INC-VGGNet.

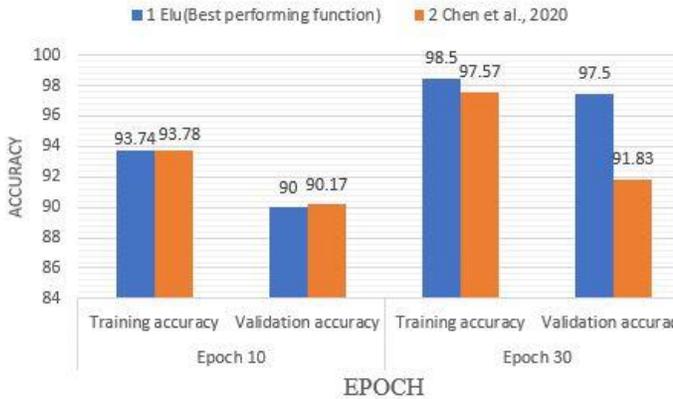


Figure 8a: Training and Validation Accuracy Chart for ELU and INC-VGGNet

This notable performance can be attributed to the inherent capability of the ELU function to effectively handle highly dynamic and complex data with nonlinear relationships. The

summary results for other performance metrics are presented in Table 2.

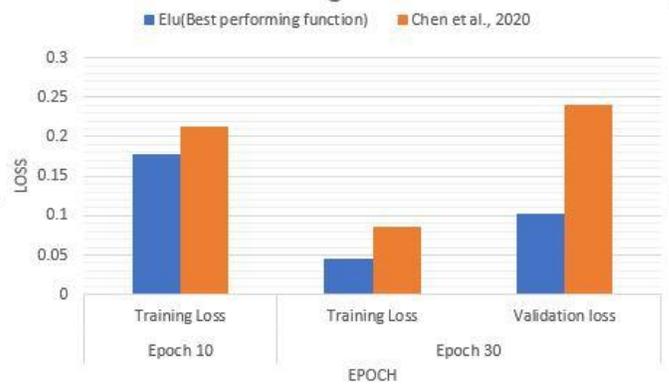


Figure 8b: Training and Validation Loss Chart for ELU and INC-VGGNet

Analysis of the table reveals that the classification accuracy for the "Healthy" class is generally lower compared to other classes, likely due to some of the preprocessed images were observed to be blurred. Furthermore, an examination of these functions' performance on the Plant Village dataset also indicated that the curated dataset used in this research exhibited superior accuracy compared to that obtained using the Plant Village dataset. This difference could be attributed to the higher quality of images in the curated dataset, as well as the preprocessing techniques employed to ensure there are no overlapping images of certain disease classes with others during the data acquisition process.

Table 2: Summary of Performance Metrics on the Curated Maize Dataset

Disease class	Specificity	Sensitivity	F1-score
Leaf blight	80.17	70.14	91.20
Common_rust	88.92	84.44	90.75
Bacterial_Streak	90.01	60.00	89.96
Healthy	60.05	79.67	49.30
Average	79.79	73.56	80.30

2) Results for plant village maize dataset

The analytical process used in analyzing the results obtained on the curated maize dataset, was also used in analyzing the results of the plant village dataset. It was also observed that the performance of the ELU activation function outperformed the other four functions tested. This indicates the ELUs activation function robustness in learning the disease classification well during the training process as evidence from the 96.45% and 91.17% training and validation accuracies recorded respectively (Figure 9).

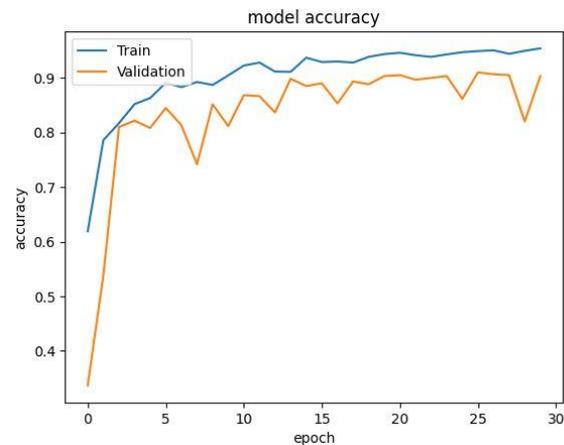


Figure 9a: Training and Validation Accuracy for ELU



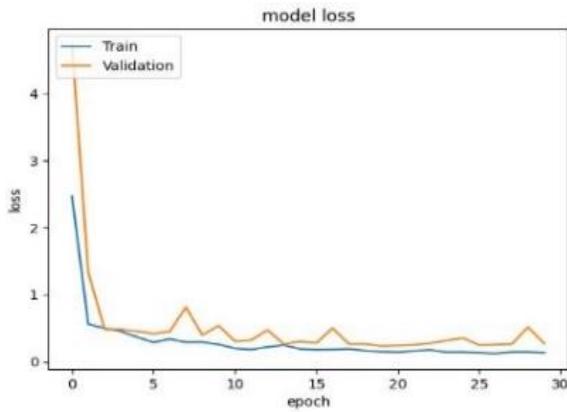


Figure 9b: Loss Graph for ELU

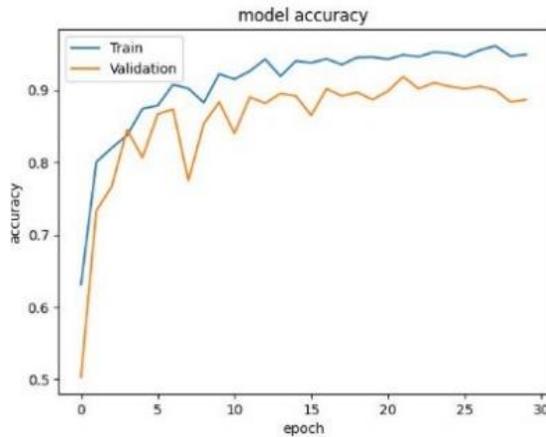


Figure 10a: Training and Validation Accuracy for Swish

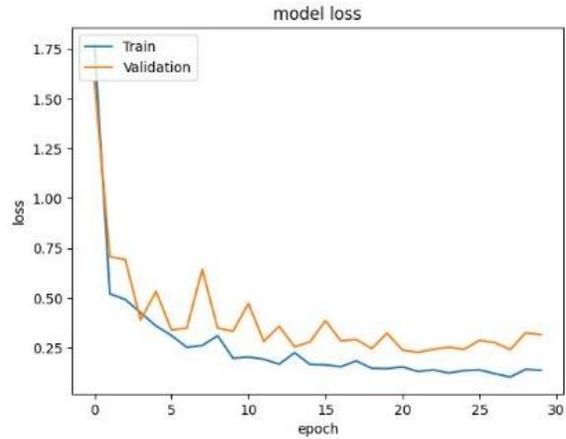


Figure 10b: Loss Graph for Swish

The graphs of ELU and swish functions performances are given in Figures 9 and 10 respectively. The performance summary of other functions can be found in Table 3.

Table 3: Summary of Different Functions Performance on Plant Village Maize Dataset

S/N	Function	Epoch 10				Epoch 30				Testing accuracy
		Training accuracy	Validation accuracy	Training Loss	Validation Loss	Training accuracy	Validation accuracy	Training Loss	Validation Loss	
1	Elu	91.55	86.950	0.2274	0.3510	96.45	91.17	0.1134	0.3070	91.56
2	LeakyReLU	91.20	85.33	0.2671	0.3101	95.00	89.83	0.1362	0.2580	90.38
3	Swish	92.20	88.33	0.1968	0.3322	94.90	88.67	0.1362	0.3157	90.75
4	PReLU	91.95	87.50	0.2400	0.2938	95.60	89.17	0.0993	0.3033	90.87
5	Selu	92.90	85.17	0.2259	0.3977	95.95	89.33	0.1124	0.3333	90.87

## V. CONCLUSION

In this study, activation functions were incorporated into a hybrid model comprising VGGNet and Inception Net, aimed at addressing the vanishing gradient problem encountered during training in deep learning models. Maize plant was chosen as focal points due to their cultivation prevalence in Northern Nigeria. The curated dataset was collected from IAR farmlands in Zaria, supplemented by the Plant Village dataset for evaluating the model's performance. Results indicate that the ELU activation function outperformed the Swish activation function, as well as three other activation functions specifically designed to mitigate the vanishing gradient problem, as described by Chen *et al.*, (2020). The model utilizing the ELU activation function achieved remarkable accuracies, with a training accuracy of 98.40% and a validation accuracy of 96.33%. Such high performance holds significant promise for

early disease detection in agricultural settings, potentially leading to increased farm productivity.

## REFERENCES

- [1] G. Yogananda, S. A. Julma, G. Madhukar, and B. Manjula, "Rice Plant Disease Classification using Transfer Learning of Deep ResNext Model," in *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*, 2023, pp. 1-5.
- [2] P. Bachhal, V. Kukreja, and S. Ahuja, "Maize disease classification using deep learning techniques: a review," in *2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT)*, 2023, pp. 259-264.
- [3] N. Nigar, H. M. Faisal, M. Umer, O. Oki, and J. Lukose, "Improving Plant Disease Classification with Deep Learning based Prediction Model using Explainable Artificial Intelligence," *IEEE Access*, 2024.
- [4] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanekaran, "Using deep transfer learning for image-based plant disease identification,"



- Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.
- [5] J. Ma, K. Du, F. Zheng, L. Zhang, Z. Gong, and Z. Sun, "A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network," *Computers and electronics in agriculture*, vol. 154, pp. 18-24, 2018.
- [6] M. Jayanthi and D. R. Shashikumar, "Cucumber disease detection using adaptively regularised kernel-based fuzzy C-means and probabilistic neural network," *International Journal of Computational Vision and Robotics*, vol. 10, pp. 385-411, 2020.
- [7] M. A. I. Aquil and W. H. W. Ishak, "Evaluation of scratch and pre-trained convolutional neural networks for the classification of Tomato plant diseases," *Int J Artif Intell*, vol. 10, pp. 467-475, 2021.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
- [9] B. Jibrin, H. Bello-Salau, I. J. Umoh, A. J. Onumanyi, A. T. Salawudeen, and B. Yahaya, "Development of hybrid automatic segmentation technique of a single leaf from overlapping leaves image," *Journal of ICT Research and Applications*, vol. 14, pp. 257-273, 2021.
- [10] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, p. 468, 2019.
- [11] M. Murray, V. Abrol, and J. Tanner, "Activation function design for deep networks: linearity and effective initialisation," *Applied and Computational Harmonic Analysis*, vol. 59, pp. 117-154, 2022.
- [12] K. Vijayaprabakaran and K. Sathiyamurthy, "Towards activation function search for long short-term model network: A differential evolution based approach," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, pp. 2637-2650, 2022.
- [13] Y. Lu, S. Gould, and T. Ajanthan, "Bidirectionally self-normalizing neural networks," *Neural Networks*, vol. 167, pp. 283-291, 2023.
- [14] G. K. Pandey and S. Srivastava, "ResNet-18 comparative analysis of various activation functions for image classification," in *2023 International Conference on Inventive Computation Technologies (ICICT)*, 2023, pp. 595-601.
- [15] B. Harshanand and A. K. Sangaiah, "Comprehensive analysis of deep learning methodology in classification of leukocytes and enhancement using swish activation units," *Mobile networks and applications*, vol. 25, pp. 2302-2320, 2020.
- [16] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.
- [17] A. A. Alkhouly, A. Mohammed, and H. A. Hefny, "Improving the performance of deep neural networks using two proposed activation functions," *IEEE Access*, vol. 9, pp. 82249-82271, 2021.
- [18] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *Advances in neural information processing systems*, vol. 30, 2017.
- [19] T. Hiroki, R. Kotani, S. Kagiwada, U. Hiroyuki, and H. Iyatomi, "Diagnosis of multiple cucumber infections with convolutional neural networks," in *2018 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2018, pp. 1-4.
- [20] M. A. Khan, T. Akram, M. Sharif, K. Javed, M. Raza, and T. Saba, "An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection," *Multimedia Tools and Applications*, pp. 1-30, 2020.

**First A. Author** (M'76–SM'81–F'87) and the other authors may include biographies at the end of regular papers. Biographies are often not included in conference-related papers. This author became a Member (M) of IEEE in 1976, a Senior Member (SM) in 1981, and a Fellow (F) in 1987. The first paragraph may contain a place and/or date of birth (list place, then date). Next, the author's educational background is listed. The degrees should be listed with type of degree in what field, which institution, city, state or country, and year degree was earned. The author's major field of study should be lower-cased.

The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.

