

Development of a transfer learning based modified VGGNet model with exponential linear unit activation function for maize disease classification

¹Abubakar, B., ²Ali, S. D., ¹Hassan, R., ¹Ahmed, A., ³Adamu, S. and ⁴Isa, S. A.

¹Department of Computer Engineering, Federal Polytechnic Bali, Taraba State, Nigeria

²Department of Information and Communication Technology, Air force Institute of Technology, Kaduna State, Nigeria

³Department of Computer Science, Federal Polytechnic Bali, Taraba State, Nigeria

⁴Department of Agricultural & Bio-Environmental Engineering, Federal Polytechnic Bali, Taraba State, Nigeria

Paper History

Received: 01st Nov., 2024

Accepted: 27th Nov., 2024

Published: November, 2024

Abstract:

Maize plant is globally grown and used as food for end consumers, and as raw material in industries for further processing. This plant is often faced with variety of diseases that affect the farm output. Different machine learning models were employed to automate the classification of these diseases where recently, Deep Learning (DL) models have shown a superior performance over other machine learning approach. But DL traditional activation function Rectified Linear Unit (ReLU) is faced with a dying neurons problem. This research curates maize dataset with four classes and modified Visual Geometry Group (VGG-19) and replaced the fifth convolutional layer with inception modules and substituted the ReLU function with Exponential Linear unit (ELU) function. The modified VGG-19 had a training accuracy of 95.52% on the curated dataset and 92.50% on a publicly available plant village dataset. The modified model had a higher performance on the curated dataset due the superior quality of images of the dataset. The modified model also outperformed the VGG-19 for two reasons, the modified architecture of the DL, and the ELU activation used which mitigate the dying neurons faced by the ReLU. This research reveals the ELU robustness in learning dynamic nature of plant disease dataset over the ReLU. Also, the curated datasets serve as pivotal resource for evaluating algorithm and model performance in maize disease detection and classification tasks, thereby advancing research in agricultural technology and fostering improvements in crop management practices. This piece of work when incorporated into Personal Digital Assistant (PDA) devices will help farmers scan through their farms for an easy identification and classification of maize diseases with higher accuracy.

Corresponding author

Abubakar, B.

abubakarbashir28@yahoo.com

Keywords: Activations, Architecture, Dataset, ELU, Exponential, Model, Plant, VGG-19

1. Introduction

The United Nations' Food and Agriculture Organization reports that plant diseases and pests is responsible for more than a third of annual natural crop losses [1]. In severe instances, plant diseases can result in complete crop failure. Therefore, it is essential to monitor plant diseases early, as this helps contain them at their initial stages by guiding appropriate treatment measures, preventing further spread within the plant or to neighboring plants [2]. Typically, plant leaves are the primary indicators of disease, with many infections showing their first symptoms on foliage [3-5]. Consequently, there exists a significant demand within the agricultural information industry for automated systems capable of detecting and identifying plant diseases [6]. Various methodologies have been developed to address this issue, with deep learning

(DL) emerging as the preferred approach due to its superior performance [3]. The use of modified pre-trained DL models is crucial in the realm of plant disease detection. These models are able to accurately identify plant diseases at an early stage and have the potential to significantly improve crop yields and reduce economic losses. This capability is vital for the agricultural sector, where timely and precise disease detection can prevent widespread crop damage and financial loss [2].

Several literatures have examined the modification of DL models in order to improve the performance of the models. For example, Hiroki, *et al.* (2018) fine-tuned the VGGNET Convolutional neural network sigmoid function as its classifier. The model considered seven viral diseases and four fungal diseases of the cucumber leaves. The model achieved an accuracy of 95.5% and was almost

nine times faster when compared to the original VGGNET architecture [7]. Also, Cap, *et al.* (2018) developed a leaf disease detection and diagnostic system. This comprised the utilization of two models namely, the CNNdetect and CNNdiag. The CNNdetect was utilized for the segmentation and detection of leaves with diseases from the aerial captured leaves images that contained many leaves in a single image. The CNNdiag was used to trigger the system to diagnose leaves with infections. Results obtained by Cap, *et al.* showed that the proposed approach, when applied on healthy leaves images was able to achieve a precision and recall of about 70% and 80.3% respectively. Although, its performance drops to about 61.5% and 55.9% for precision and recalls when the model was deployed for use in the field. However, there is need to investigate approaches that can help improve the training time and performance accuracy of the models [7].

Zhang, *et al.* (2019) proposed a model Global Pooling Dilated Convolutional Neural Network (GPDCNN) for cucumber disease classification, where AlexNet was edited and fine-tuned, the first convolution layer was substituted with the dilated convolution kernel, after the fourth pooling layer, an Inception and feature fusion (Concat) layers were added. The two fully connected layers were also substituted with a global pooling layer after the fifth convolution and finally the softmax classifier. The model outperformed DCNN and the original AlexNet architecture with an accuracy of 94.65% as compared to the 91.73% and 92.48% for DCNN and AlexNet respectively [8].

Despite the advancements in pre-trained DL models, there is a notable lack of research focusing on how different activation functions affect their performance in plant disease detection. Activation functions play a critical role in the learning process of neural networks, influencing the model's ability to capture complex patterns in data [9]. Understanding the activation function impact is essential for optimizing model performance and achieving higher accuracy in disease detection. This research curates maize plant disease dataset and investigates the effect of ELU activation function in modified pre-trained VGG-19 DL architecture.

1.1 Fundamental Concepts

This section has three parts, where the first sections briefly discuss concept of the convolutional neural networks (CNN) and DL, the second section is activation functions and their variants. Finally, the third section presents the concept of transfer learning and DL pre-trained models.

1.1.1 CNN and Deep Learning

A CNN is a neural network that incorporates the convolution operation in at least one layer. A convolution operation involves performing a dot-product between a grid of weights and corresponding grid-structured inputs that are sampled from various spatial locations within the input volume [10]. This method is particularly effective for data characterized by significant spatial or local relationships, such as images [11]. After convolution, there is a pooling

layer, where spatial sizes of the images are reduced by one of these methods; maxpooling, minpooling or random pooling. The pooling operation operates on small grid regions of size $P_q \times P_q$ in each layer, generating a new layer that maintains the same depth (unlike filters). In contrast to convolution operations, pooling is performed on each activation map individually. This means that pooling works independently on each feature map to create another feature map. As a result, the pooling operation does not alter the number of feature maps [11]. The output of the pooling layer is passed through the traditional activation function of CNN known as the rectified linear unit (ReLU) [11].

Deep learning is a network that has several layers of artificial neural network. DL models require huge data for training, in a scenario where the data is small, it will be better to enhance the shallow machine learning models where the result will even be easier to translate [8].

1.1.2 Activation functions

In the training of deep learning networks especially those networks that have large number of layers, a common problem encountered is the gradients becoming too small during the training, this is called the vanishing gradient. This affects the learning rate as the weights of the layers are updated slowly during the back propagation and this leads to the network not learning significant features from the data [9]. In addressing this problem, many activation functions were proposed and the Rectified Linear Unit (ReLU) performed remarkably well in solving the vanishing gradient problem [12]. Nevertheless, even with its performance there are functions proposed which perform better than the ReLU in mitigating the dying neurons problem which is common in the ReLU, these include the Leaky Rectified Linear Unit (LeakyReLU) [13], Swish [12], Exponential Linear Unit (ELU) [14], Scaled Exponential Linear Unit (SELU) [15], and the Parametric Rectified Linear Unit (PReLU) [16]. Equations 1-5 represent the ReLU, LeakyReLU, Swish, ELU, SELU and PReLU respectively [12-16].

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

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

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

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

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

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

1.1.3 Transfer learning and DL models

Transfer learning is defined as the method where CNN trained for a particular job is used again as the preliminary point for a model for a second task. Weights of a pre-trained model can be used for training large labeled dataset instead of randomizing the initial weights [3, 5].

AlexNet won the “ImageNet Large Scale Visual Recognition Challenge (ILSVRC)” 2012. In its internal architecture, it has two corresponding pipelines of preprocessing; makes two GPUs working collectively to build a training model faster in speed and memory sharing. AlexNet accept an input of size 224x224x3 and has 96 filters with size 11x11x3 for the first layer. With 4 strides, the first layer result is a 55x55x96, then max-pooling. Every convolution layer will be applied with a ReLU function, then response normalization follows and then the max-pooling. As various convolution layers have varying kernel sizes, all the pooling layers have a 3x3 kernel with 2 strides. The fully connected layers have 4096 neurons, which have 4096-dimension denoting the representation of the input image. The final layer of the AlexNet is the softmax layer which has a 1000-way classification [11]

VGG19 and VGG16 are the two typical models of the VGGNet pre-trained models with 19 and 16 layers respectively trained on ImageNet dataset. The VGG has a cascaded network structure, having 2 to 3 convolutional layers which pooling layer follows. The convolutional layers have small layers of 3x3 convolutional filters [17].

Novel inception architecture was proposed in the GoogleNet, which is network inside a network. Since then, the inception architecture has received much attention by many researchers. Important information of image is found at various layers of the inception, which make it more scalable, and this is the fundamental key to its performance. The GoogleNet has 22 different inception layers of different filter sizes and average pooling replaced fully connected layer in its architecture. Despite been able to win in the 2014 ILSVRC with error rate of 6.7%, GoogleNet architecture is complex as compared to the runner up VGGNET [18].

2. Materials and methods

This section has three parts, the first part describes the dataset developed and used for training and testing of this research work, the second part describes the architecture of the Modified VGG19 adopted for the work. The last part gives details of the parameters and hyperparameter settings of the models.

2.1 Dataset

Two datasets were used in this study. The first dataset is the maize dataset curated in Bali Axis for the purpose of this research. The second dataset is the open-source plant village maize dataset which can be found in www.kaggle.com. After consultation with plants disease experts from the Institute of Agricultural research Zaria, some of the maize diseases of the publicly available dataset were said to be mixed and some wrongly classified. With guidance of the experts at crop protection of the IAR, a new plant disease dataset was curated for the maize plants considered in this study. A digital camera with model Canon 100 was used for the data capturing of plant diseases images on the farms within the Federal Polytechnic Bali. Four classes of maize diseases were captured, namely northern leaf blight, bacterial leaf Streak, common-rust, and the healthy maize. It was ensured that every class has at least 100 images each, after that the images were then preprocessed, including background removal, sharpening and image resizing (224x224). The images were then augmented using various augmentation techniques, such as rotation, vertical and horizontal flip, shear, zoom, among others. After augmenting each of the class images, the resultant image of each class were 500 images. The samples of preprocessed images are shown in Figure 1.

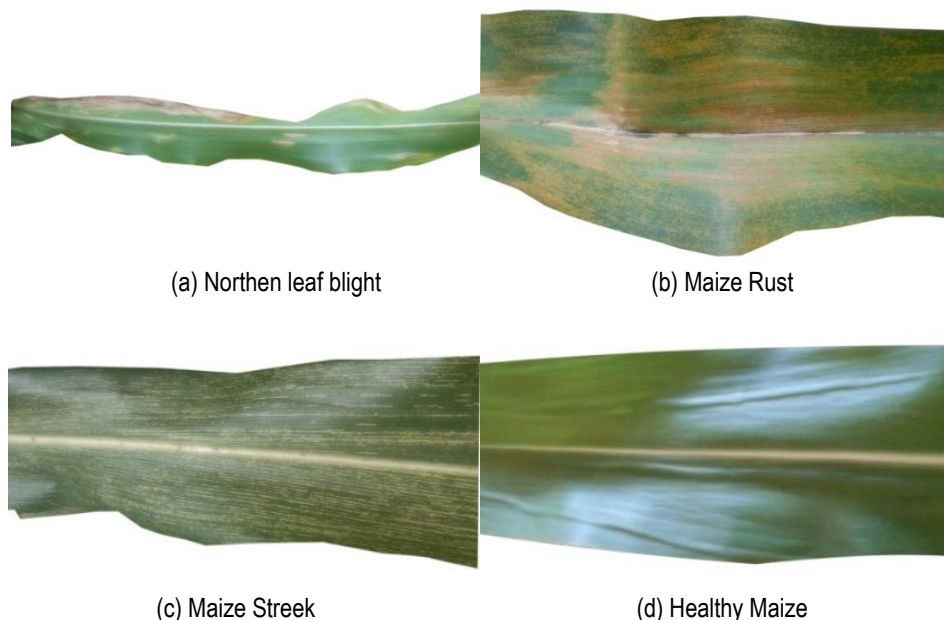


Figure 1: Preprocessed sample images of the curated maize dataset

2.2 Dataset Division and Performance Metrics

The dataset collected for this research work include four classes of maize which include the Northern Leaf

Blight, Maize Rust, Maize Streak and Healthy class. These images were split into training, validation and testing sets. The training and validation for the dataset has at least

2000 and 600 images for the four classes respectively, while the testing set also has 600 images. This gives the total number of images for the dataset to be at least to be 3200. The training and testing ratio is 70/30% from the numbers split.

The performance of the models is evaluated using metrics such as accuracy, precision, F1-score and Recall. These metrics were usually computed using the relationship between True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN). The TP is the correctly classify sample as positive which is actually positive, TN is the correctly classify sample as negative, which is actually negative, FP is the classification of a sample as positive which is actually negative, FN predict sample as negative while actual classification should be positive.

The precision (Pr) can simply be described as the ratio of the correctly predicted observed samples (TP) to the total predicted positive observed samples, which comprises of the misclassify samples as positive (FP) and the TP as shown equation (1) [19]. Similarly, the Recall (Rc) is the ratio of the correctly predicted observed samples to the total observed samples as expressed equation (2) [20]. Also, the accuracy (A) is evaluated as the ratio of the correctly classify samples to the total observed classes (top) as expressed in equation (3). The F1-Score ($F1$) is the harmonic mean of the precision and the recall rate and usually computed using equation (4) [21].

$$Pr = \frac{TP}{TP+FP} \quad (1)$$

$$Rc = \frac{TP}{TP+FN} \quad (2)$$

$$A = \frac{TP+TN}{top} \quad (3)$$

$$F1 = 2 \times \frac{Pr \times Rc}{Pr+Rc} \quad (4)$$

2.3 Modification of VGG19 Architecture

The VGG19 fifth convolutional layer and other layers beyond were removed and replaced with convolutional layer of dimension 3x3x512 and an ELU activation function was also added. The ELU function was chosen because it outperformed SELU, PReLU, Swish, or LeakyReLU functions on the maize dataset after rigorous training and testing with the individual functions. After that, two inception layers were added with each having a feature map of 256, the global pooling layer and Softmax activation functions were added as the top layers. The new generated network after compilation had approximately 13.4 million parameters of which only about 2.7 million are trainable. The trainable parameters are the weights of the network which are updated to learn the features and adjusted to suit the task at hand. To lower the errors between the true labels and the predicted labels, these weights are adjusted during the training process using backpropagation. The non-trainable parameters are the weights that are frozen and are not updated during the training but rather adopted from the earlier pre-trained

model. This is due to the fact that mostly the computational resources and dataset at hand usually are very limited compared to the originally pre-trained computational resources and dataset. The adoption of the non-learnable parameters also helps in lowering the chances of overfitting of the trained model. The learnable parameters are from the layers added while non-learnable parameters are from the earlier pre-trained VGGNet architecture layers adopted. Flowchart of the newly developed architecture after modification is given in Figure 2.

3. Results and Discussion

This section comprises of three parts. The first part presents the results and discussion of the model's performance on the collected dataset of this study, while the second section presents the results and discussion of other performance metrics of the modified model's performance on collected dataset. The third section presents results and discussion of the model's performance on the plant village dataset.

3.1 Models on Collected Maize Dataset

The training Accuracy and training Loss of a model determines how well a model is able to approximate or represent a classification problem. The training and validation accuracy is 95.40% and 91.83% respectively (fig. 3a), while the training and validation loss of the modified VGGNet is 0.2672 and 0.3579 respectively (see fig. 3b). The training and validation accuracy of VGG-19 is 62.50% and 79.17% respectively as presented in Fig. 4a, while training and validation loss of the model on the collected dataset is 1.2049 and 1.0984 respectively as shown in Figure 4b.

The higher Accuracy and lower Loss values of the modified architecture of this research indicates that the model after training was able represent the classification data closely to the expected output compared to the original VGG-19. This is attributed to two reasons. First is the difference between number of parameters of the modified model with 10 million parameters while VGG-19 has 155 million approximately. This reduces the complexity of the modified model while training on a new dataset, leading to lower training time and computational resources needed. Second reason the modified model outperformed the VGG-19 is the ELU activation function that solves the dying neurons problem faced by the traditional ReLU function of the CNNs. The ELU is able to handle the non-linearity and dynamic nature relationship of the maize disease dataset better than the ReLU. This is revealed in the training and validation graph where the model steadily improves as the epochs increases. Similarly, the difference between the training and validation Accuracy of the VGG-19 model is much indicating that the model overfitted due to the smaller number of images in the dataset compared to the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). The model finds it difficult to learn the dynamic and non-linear relationships in the dataset. The summary of the performances of the two models on both the curated and plant village dataset is given in Table 1.

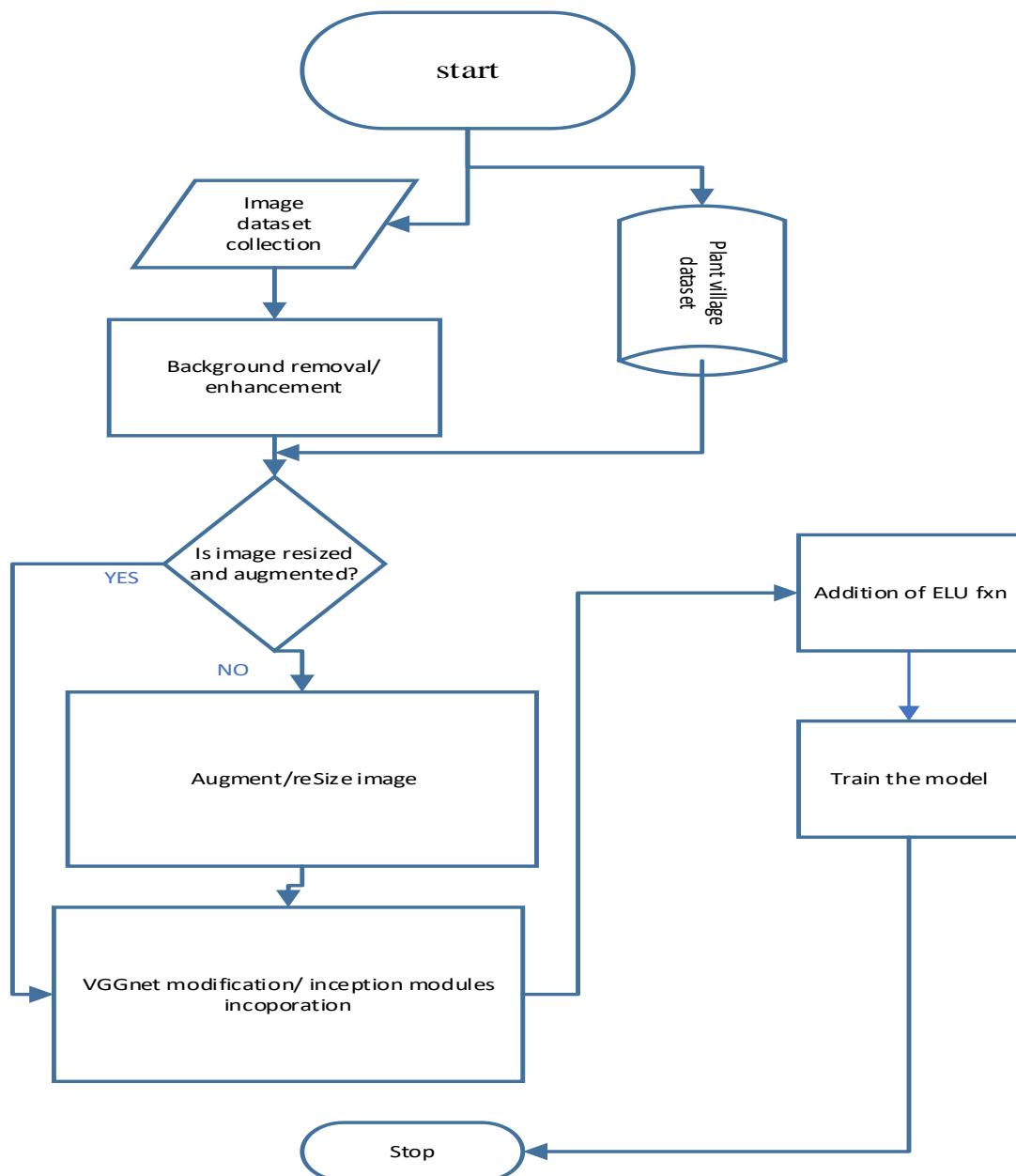
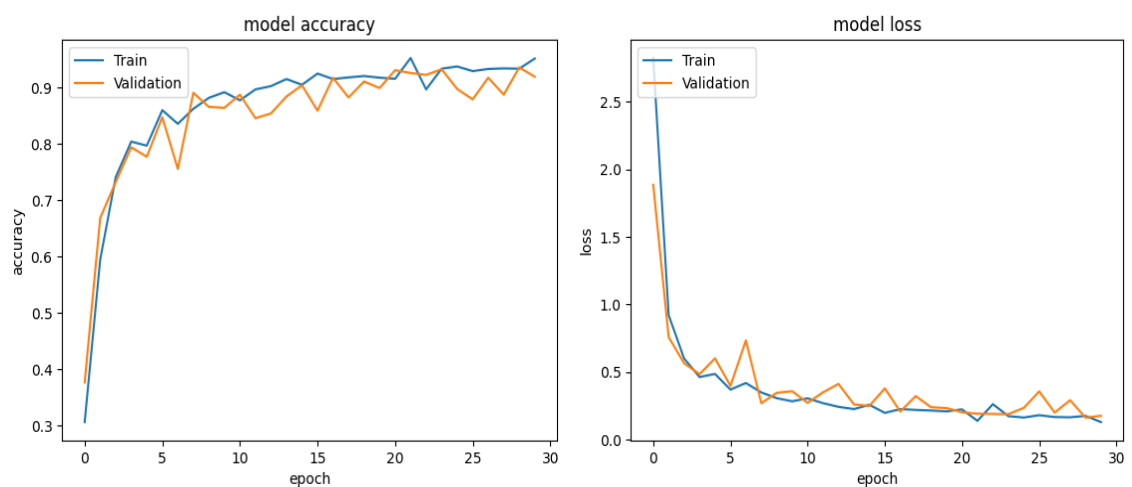
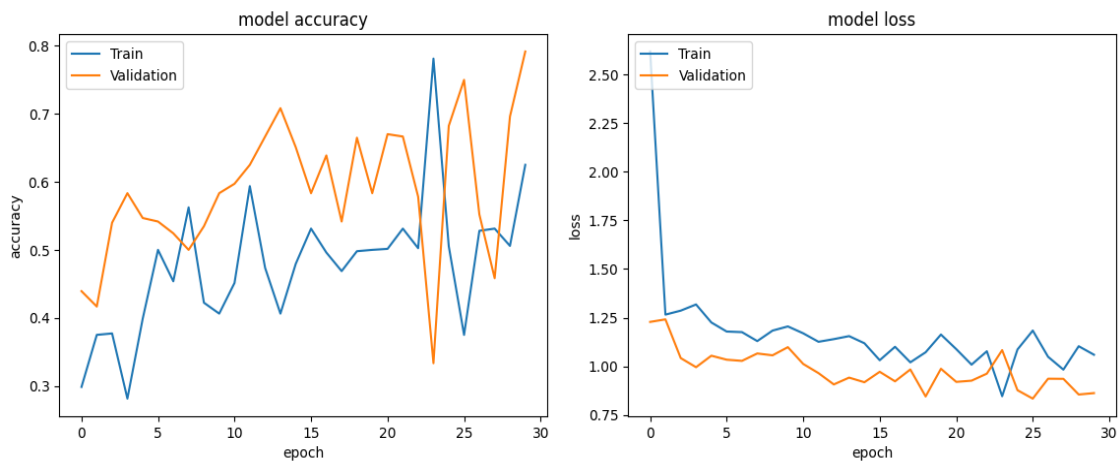


Figure 2: The modified VGG19 flowchart



(a) Accuracy
(b) Loss
Figure 3: Modified VGGNet training accuracy and loss graphs for collected dataset



(a) Accuracy (b) Loss
Figure 4: VGG-19 Training accuracy and loss graphs for collected dataset

Table 1: Summary of performances of the models on the datasets

S/N	Model	Dataset	Epoch 10				Epoch 30				Testing
			Training accuracy	Validation accuracy	Training loss	Validation loss	Training accuracy	Validation accuracy	Training loss	Validation loss	
1	Modified	Curated	89.81	86.33	0.2672	0.3579	95.40	91.83	0.1298	0.1754	89.6
2	Vgg-19	Curated	78.12	79.17	0.5713	0.5277	68.75	78.17	0.5479	0.4794	56.77
3	Vgg-19	Plant village	40.62	58.33	1.2049	1.0984	62.50	79.17	1.0594	0.8624	49.99
4	Modified	Plant village	88.74	83.17	0.2683	0.3972	92.52	90.33	0.1895	0.2827	87.65

3.2 Other Performance metrics of the model

Table 2 gives a summary of the modified model's performance on precision, sensitivity, specificity and F1-Score. Analysis of the table reveals that the classification accuracy for the "Healthy" class is generally lower compared to other classes, likely due to some preprocessed images were observed to be blurred.

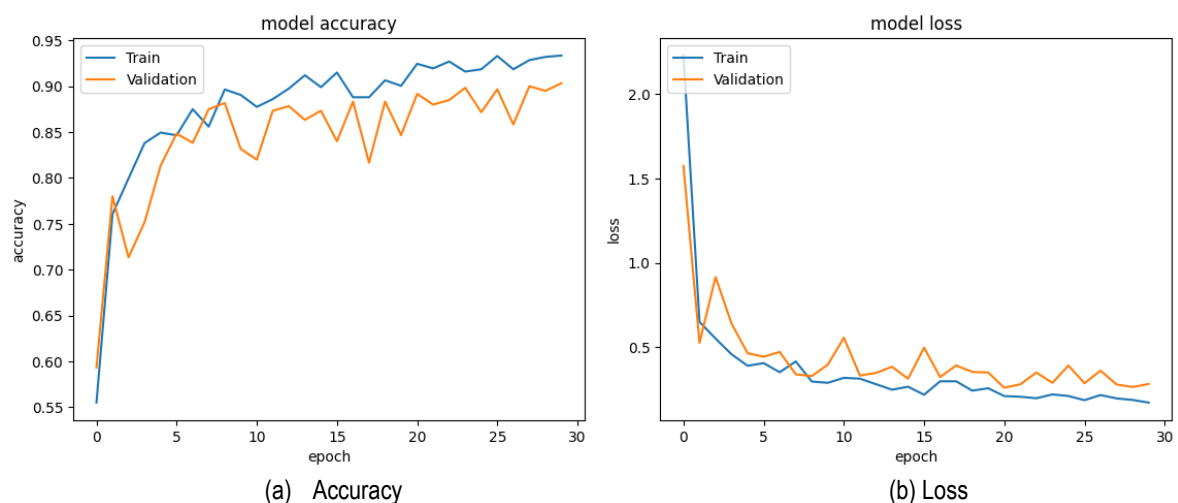
Table 2: Summary of performance metrics on the collected maize dataset

Disease class	Specificity	Sensitivity	F1-score
Leaf blight	78.86	68.15	90.55
Common_rust	85.90	82.67	89.65
Bacterial_Streak	88.40	60.50	86.56
Healthy	58.50	69.77	53.50
Average	77.92	70.27	80.07

Furthermore, an examination of the 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.

3.3 Models on Plant Village Maize Dataset

The analysis of the models' performance on the plant village maize dataset follows the same suit as the analysis of the collected dataset. The dataset also has four classes, namely; Blight, Common_Rust, Gray_leaf_spot, and healthy. The training and validation accuracy of the modified model is 92.52% and 90.33% respectively as shown in Figure 4a at epoch 30. While the training and validation Loss is 0.1895 and 0.2827 respectively as shown in Figure 4b at epoch 30.



(a) Accuracy (b) Loss
Figure 4: Modified VGGNet accuracy and loss graphs for plant village dataset

The training and validation accuracy firmly increases which shows the models' robustness and ability to learn and relearn steadily at each epoch.

Despite the fact that the model performs greatly on the plant village dataset, it is less compared to its performance on the collected dataset. This can be attributed to the superior quality of images captured during the dataset collection which influences the model's performance. Likewise the preprocessing techniques employed on the images. The preprocessing techniques

includes including background removal, image sharpening, and different forms of image augmentation.

The training and validation accuracy of VGG-19 model as shown in Figure 5a is 68.75% and 78.17% respectively at epoch 30, while Figure 5b shows the training and validation Loss is 0.5479 and 0.4794 respectively at epoch 30. The validation accuracy been higher than the training accuracy indicates a sign of overfitting of the model. The smaller size of the dataset compared to the ILSVC dataset on which the VGG-19 was initially trained on, could be the reason of the overfitting.

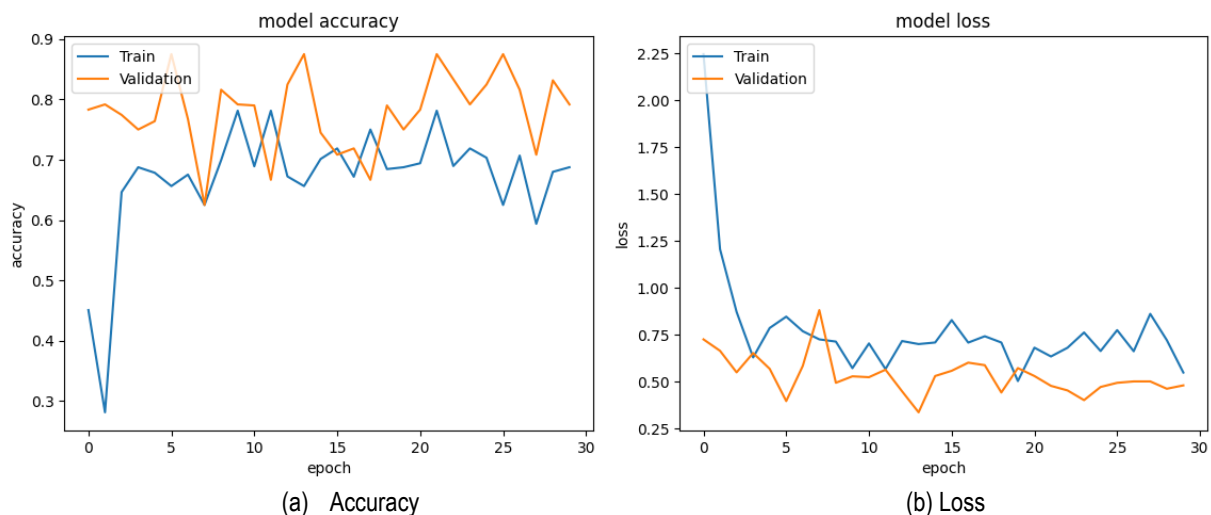


Figure 5: VGG-19 Training Accuracy and Loss graphs for Plant Village Dataset

4. Conclusion

This research work curated a maize dataset and modified the VGG-19 pre-trained model. The fifth convolutional layer of VGG-19 was replaced with two inception modules each with an ELU activation function. The modified architecture when trained outperformed the VGG-19 on the both curated dataset of this work and the plant village dataset which is publicly available at www.kaggle.com. The modified model achieved a remarkable accuracy of 95.40% at 30 epochs on the collected dataset and 92.52%. The higher performance on the collected dataset is attributed the superior quality of the images over the plant village dataset and non-overlapping of multiple diseases on a plant disease.

Recommendation

There is a need for creation of an enhanced grain-based disease dataset, focusing particularly on diseases that are challenging to culture in laboratory settings. Also, it is recommended in further works to examining how the timing of training and classification impacts the performance of different activation functions within the proposed model for disease classification.

Acknowledgement

This work is supported by the Tertiary Education Trust Fund (TETFUND) under the Institution Based Research (IBR) with ID. (TETF/DR&D/CE/POLY/BALI/TARABA/IBR/2022/VOL.1).

The Author would also like to express his gratitude to the Research and Development committee of Federal Polytechnic Bali for their insightful comments and ideas that greatly contributed to the development of this study.

Reference

- [1]. Cap, Q., Suwa, K., Fujita, E., Uga, H., Kagiwada, S. and Iyatomi, H., (2018). An end-to-end practical plant disease diagnosis system for wide-angle cucumber images, *International Journal of Engineering & Technology*, 7, 106-111.
- [2]. Aquil M. A. I. and Ishak, W. H. W., (2021). Evaluation of scratch and pre-trained convolutional neural networks for the classification of Tomato plant diseases, *International Journal of Artificial Intelligence*, 10, 467-475.
- [3]. Chen, J., Chen, J., Zhang, D., Sun, Y. and Nanehkaran, Y. A., (2020). Using deep transfer learning for image-based plant disease identification, *Computers and Electronics in Agriculture*, 173, 105393.
- [4]. Yadav, S., Sengar, N., Singh, A., Singh, A. and Dutta, M. K. (2021). Identification of disease using deep learning and evaluation of bacteriosis in peach leaf, *Ecological Informatics*, 61, 101247.
- [5]. Gao, Z., Luo Z., Zhang W., Lv, Z. and Xu, Y., (2020). Deep learning application in plant stress imaging: a review, *AgriEngineering*, 2, 430-446.
- [6]. Xu, J., Lowe, C., Hernandez-Leon, S. G., Dreisigacker, S., Reynolds, M. P., Valenzuela-Soto,

- E. M., and Heuer, S. (2022). The effects of brief heat during early booting on reproductive, developmental, and chlorophyll physiological performance in common wheat (*Triticum aestivum* L.). *Frontiers in Plant Science*, 13, 886541.
- [7]. Hiroki, T., Kotani, R., Kagiwada, S., Hiroyuki, U. and Iyatomi, H., (2018). Diagnosis of multiple cucumber infections with convolutional neural networks, *IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, Washington DC, USA, pp. 1-4.
- [8]. Zhang, S., Huang, W. and Zhang, C., (2019). Three-channel convolutional neural networks for vegetable leaf disease recognition, *Cognitive Systems Research*, 53, 31-41.
- [9]. Pandey, G. K., & Srivastava, S. (2023). *ResNet-18 comparative analysis of various activation functions for image classification*, In 2023 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal. pp. 595-601.
- [10]. Taye, M. M. (2023). Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation*, 11(3), pp. 52-60.
- [11]. Aggarwal, C. C. (2018). *Neural networks and deep learning*, Vol. 10, No. 978, p. 3. Cham springer.
- [12]. Harshanand B. and Sangaiah, A. K., (2020). Comprehensive analysis of deep learning methodology in classification of leukocytes and enhancement using swish activation units, *Mobile networks and applications*, 25, 2302-2320.
- [13]. Nair V. and Hinton, G. E., (2010). *Rectified linear units improve restricted boltzmann machines*, Proceedings of the 27th international conference on machine learning (ICML-10), Canada, pp. 807-814.
- [14]. Alkhouly, A. A., Mohammed, A. and Hefny, H. A., (2021). Improving the performance of deep neural networks using two proposed activation functions, *IEEE Access*, 9, 82249-82271.
- [15]. Klambauer, G., Unterthiner, T., Mayr, A. and Hochreiter, S., (2017). Self-normalizing neural networks, *Advances in neural information processing systems*, in 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA pp1-30.
- [16]. He, K., Zhang, X., Ren, S. and Sun, J., (2016). *Deep residual learning for image recognition*, Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, Nevada, pp. 770-778.
- [17]. Phine, S., (2023). Pneumonia classification using deep learning vgg19 model, in *IEEE Conference on Computer Applications*, (ICCA), Yangon, Myanmar, pp. 67-71.
- [18]. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S. and Anguelov, D., (2015). Going deeper with convolutions, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Boston, pp. 1-9.
- [19]. Powers, D. M., (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 2229-3981.
- [20]. Sokolova, M. and Lapalme, G., (2009). A systematic analysis of performance measures for classification tasks, *Information processing and management*, 45, 427-437.
- [21]. Owusu-Adjei, M., Ben Hayfron-Acquah, J., Frimpong, T. and Abdul-Salaam, G., (2023). Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems, *PLOS Digital Health*, 2, e0000290.