The Open Agriculture Journal
DOI: 10.2174/0118743315288623240223072349, 2024, 18, e18743315288623
ISSN: 1874-3315

RESEARCH ARTICLE
OPEN ACCESS

Intelligent Plant Leaf Disease Detection Using Generative Adversarial Networks: a Case-study of Cassava Leaves

Gururaj Harinahalli Lokesh[1,*], Soundarya Bidare Chandregowda[6], Janhavi Vishwanath[2], Vinayakumar Ravi[3,*], Pradeep Ravi[4] and Alanoud Al Mazroa[5]

[1]*Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India*
[2]*Department of Computer Science and Engineering, Vidyavardhaka College of Engineering, Mysuru, Karnataka, India*
[3]*Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia*
[4]*Department of Information Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, Karnataka, India*
[5]*Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), P.O. Box 84428, Riyadh 11671, Saudi Arabia*
[6]*Department of Artificial Intelligence and Machine Learning, Alva's Institute of Engineering and Technology, Mangalore, India*

Abstract:

Background: Cassava leaf disease detection is a major problem since it is very difficult to identify the disease in naked-eye observation and even experts such as agricultural scientists can fail in this task. The approach we use in this paper has the potential to overcome this problem.

Materials and Methods: In this, we propose an approach based on artificial intelligence for leaf disease detection using deep learning with generative adversarial networks (GAN). Our experimental study used a dataset including 12880 cassava leaf pictures generated using CycleGAN showing five major disease classes. In order to avoid overfitting, a GAN architecture is proposed for data augmentation using two networks, *i.e.*, a Generator and a Discriminator. The generator is trained to generate similar data samples as the original data

Results: The proposed approach achieved an accuracy of 99.51% for the classification of healthy or unhealthy leaf images, which outperformed existing methods.

Discussion: The discriminator is trained to distinguish between the unique and generated sample records, as actual or fake. To classify cassava images into five categories of diseases, a combination of machine learning models has been trained on original and generated images. The proposed approach showed better accuracy compared to the existing methods.

Conclusion: The proposed deep learning-based method can be used as a tool for early disease diagnosis in cassava leaf disease detection and classification

Keywords: Plant leaf disease, Cassava, Deep learning, CycleGAN, CNN, Deep learning.

*Address correspondence to these authors at the Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Khobar, Saudi Arabia and Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India
E-mails: gururaj.hl@manipal.edu and vinayakumarr77@gmail.com

Cite as: Lokesh G, Chandregowda S, Vishwanath J, Ravi V, Ravi P, Al Mazroa A. Intelligent Plant Leaf Disease Detection Using Generative Adversarial Networks: a Case-study of Cassava Leaves. Open Agric J, 2024; 18: e18743315288623. http://dx.doi.org/10.2174/0118743315288623240223072349

Received: November 26, 2023
Revised: January 23, 2024
Accepted: February 02, 2024
Published: March 05, 2024

Send Orders for Reprints to
reprints@benthamscience.net

1. INTRODUCTION

Cassava is the primary crop in Africa and many other international locations. Africa is one of the biggest cultivators of Cassava plants. Poor quality cultivation can lead to diseases [1], which are similar to maize leaf sicknesses [2]. The early detection of leaf disease allows cultivars to be saved before they become permanently affected [3]. Previous research focused on an automated system to detect and count agricultural disease formation from leaf images [4]. Leaves that can be attacked employing the disorder will affect crop yields due to the fact leaves are a critical part of the plant since the phloem tissue transports the effects of photosynthesis to all special components of the plant [5].

To detect illness in cassava leaves, a laboratory or aid from a plant expert is usually used. However, laboratory testing is typically limited to employing value concerns, and an expert cannot discover it in time, so farmers are unable to rapidly and properly manage disease problems in cassava leaves. Deep learning (DL) and machine learning (ML) approaches for detecting plant leaf disease have been addressed in previous research providing promising results [6]. A smart classifier, which is connected to portable devices configured for specific diseases can improve detection accuracy by combining datasets [7]. In realistic settings, portable electronics such as smartphones, drones, and computers can be used [7].

In this work, DL and GAN are used to recognize cassava plant diseases using leaf images. In the proposed technique, we generated artificial pictures through the usage of Cycle GAN and augmented the dataset with these images to improve training. A combination of Convolutional Neural Network (CNN), VGG16, and ResNet34 models become skilled in cassava plant pictures and artificial snapshots generated by using the Cycle GAN model for the detection of sickness from the leaf pictures. This study affords a complete studying strategy for cassava leaf disease identification based on combinational approaches (CNN+VGG16+ResNet34).

The relevant original contributions of this research are:

(1) Comprehensive dataset generation of pictures of cassava leaves affected by known diseases.

(2) Analysis of how combining CNN, VGG16, and the ResNet34 model can improve cassava leaf disorder identification.

(3) Use of separable convolution reduces computational costs.

2. LITERATURE SURVEY

Cassava leaf disease prediction using ResNet-50- and SVM model is presented in reference [8]. In the first phase, the model starts by extracting all the relevant features and in the second phase classifies the image using an SVM classifier. The results show higher accuracy and overall performance with the aid of incorporating ResNet-50 and SVM classifiers. A hybrid transfer learning method is used in a digital image processing model [9]. In leaf disease research, accurate data processing is critical. This enhances the detection of plant disease patterns, forecasting, and model performance. A study [10] presents a machine-learning-based SVM and Naive Bayes model for detecting plant illnesses. Another study [11] used ImageNet to develop a deep-learning-based model for detecting Cassava disease. To classify leaf diseases, the SVM classification approach was used. In a study [12], a form and texture-based classification system for identifying Cassava disease is discussed. The proposed model had an accuracy of more than 84% and a detection rate of more than 88%. To forecast leaf disease, researchers [13] developed a deep-learning-based model. To identify thirteen specific crop diseases, this study employs a feature selection method. Using the Caffe deep-learning technique, researchers were able to train CNN architecture. To recognize fungal leaf infection, this program used a feature selection technique. In comparison to an existing model, the proposed model attained a 94.1 accuracy rate. To characterize Cassava illnesses in plants [14], presents a nine-layer-based convolutional neural network model. In small data scenarios for image recognition, data augmentation approaches have been critical in extending the training dataset. Horizontal/vertical flipping or mirroring, rotations, cropping, and other data augmentation techniques are some of the most employed. Table **1** compares and contrasts various existing approaches for detecting and analyzing plant leaf diseases.

Table 1. Comparison of Cassava leaf disease classification models.

Type	Classification	Advantages of the Proposed Method	References
Apple disease	Faster CNNs	Higher accuracy and faster detection speed	[15]
Tomato disease	CNN	Better accuracy	[16]
Five crops diseases	CNN architecture	the multi-crop diseases can be classified	[17]
Rice	KNN, backpropagation, Naïve Bayes and multiclass SVM	Better accuracy	[18]
Multiple plant species	ResNet	lowers computational cost	[19]
Tomato	Inception-ResNetv2, Autoencoder	Better accuracy	[20]
Cassava	CNN model	Lower computational cost	[21]
Cassava	Random Forest, SVM, and SCNN	Better accuracy	[22]

3. METHODOLOGY

3.1. Cassava Dataset

There are five specific classes as shown below. Each of which has a unique function that may be noticed at the leaf as shown in Fig. (**1**). CBB is a bacterial ailment, the main motive is moisture. Black leaf patches and blights are common symptoms. CBSD is caused by the Bemisia tabaci whitefly virus. There are indicators of leaves, as well as yellow chlorosis on secondary and tertiary leaf bones [23], and chlorotic patches constitute the second type of symptom. White patches on the leaves are caused by CGM. It starts as tiny spots that spread to cover the entire leaf floor, resulting in a lack of chlorophyll and thus impairing photosynthesis. CGM is caused by Mononychellus tanajoa, a bug that feeds on the undersides of younger leaves. CMD Spots of green are always present on leaves, along with the distinct coloration of yellow and white patches [24].

3.2. Architecture

Fig. (**2**) shows the architectural diagram of the proposed model. First, load the TFRecord files of different classes. Here CycleGAN is used for data augmentation. The hybrid model (CNN+VGG16+Resnet34) was built to train the generated images. The hybrid model was trained to classify whether the input image is healthy or diseased.

CNN is a completely famous approach in deep learning wherein a couple of layers are robustly trained. It has been discovered to be pretty powerful and is moreover the most broadly applied in numerous programs of computer vision. CNN operates on unorganized picture inputs and transforms them into the right output classes for type. In our paintings, a shape has been built for this set of rules. This form is crafted from several layers as shown in Fig. (**3**), which illustrates the structure that we used to construct the CNN.

By way of constructing the characteristic maps of all filters alongside the vertical height size, the outcome length of the conv layer can be obtained. The output of this layer in CNNs can be written as Eq. (**1**),

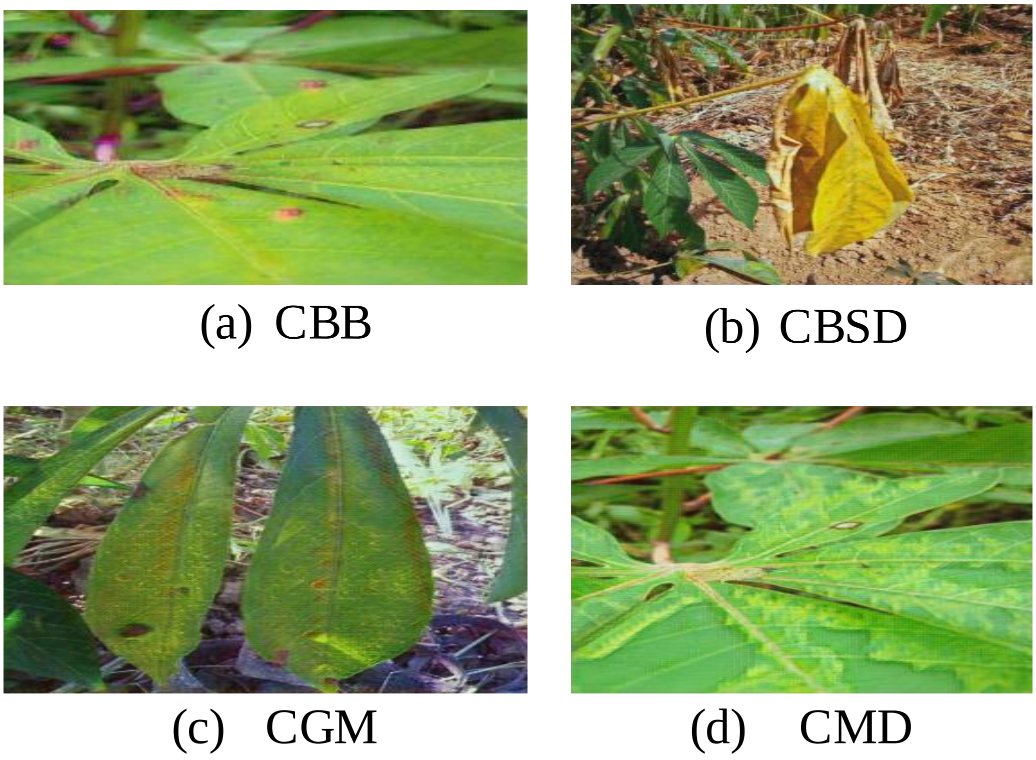
$$X_j{}^p = \sum_{\mathrm{i} \in \mathrm{N_j}} X_i^{p-1} x k_{i j^{p+B_j{}^p}} \qquad \textbf{(1)}$$

where $\mathrm{p^{th}}$ layer is represented by p, Nj denotes the number of filters, $X_{i^{p-1}}$ denotes feature map, Kij denotes convolutional kernel, and Bj denotes bias term.

The output of the final convolutional might be exceeded as entering to this accretion. It system the input for higher learning and integrates the spatial facts of the training facts.

$$\mathrm{F(x)} = \mathrm{relu}(\mathrm{x}w^T + \mathrm{b}) \qquad \textbf{(2)}$$

where the input data is presented by x, w denotes the weight vector, and the bias term is represented by b.



(a) CBB

(b) CBSD

(c) CGM

(d) CMD

Fig. (1a-d). Different categories of diseases.

The output layer is the layer that has complete connectivity with the previous layer and receives input from it. It makes use of softmax activation for predicting the target output (elegance) with excessive opportunity cost. The softmax can be mathematically written as,

$$S(z^{i)} = \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}} \; for\, \mathrm{i} \, = \, 1, \ldots, \mathrm{n} \qquad (3)$$

Where the number of classes is represented by n, the value of (*Zi*) is always a positive value within the range of (0,1). The numerator value is the input to the denominator and added with other positive numbers, the numerator takes any real value from the input vector. The value 1.0 can be summed up to output probability values.

VGG16 only contains convolutional and pooling layers, and it always uses a 3 x 3 Kernel for convolution and a 2x2 Kernel for a max pool as shown in Fig. (**4**). It has over 138 million variables. ImageNet data will be used to train it.

3.3. CycleGAN

Traditional convolutional layers are used in GAN to shape a picture matrix made of random noise. The generator's job is to create phony photos, while the discriminator's job is to tell the difference between fake and real photos. Two versions will be trained at the same time and will compete against each other. The discriminator makes sure that the images generated by the generator are as real as possible. An input layer, an embedding layer, and a dense layer are included in the CycleGAN discriminator version, along with an embedding layer, reshaping layer, and concatenate layer, which is observed by four convolutional layers. A Leaky ReLU layer is used to observe each convolutional layer. A flattened layer, a dropout layer, and a dense layer are used to follow the ultimate Leaky ReLU layer.

3.4. GUI

It is well known that farmers are not familiar with programming languages. So, it is very difficult for them to understand/analyze what is happening. Building user-friendly software helps them to identify the disease easily. It can be operated easily by way of a beginner. The user interface of the cell utility includes the handiest two buttons. Initially, a user has to upload the image to be classified. and the second button is assessed, which functions to classify the input picture using the hybrid model.

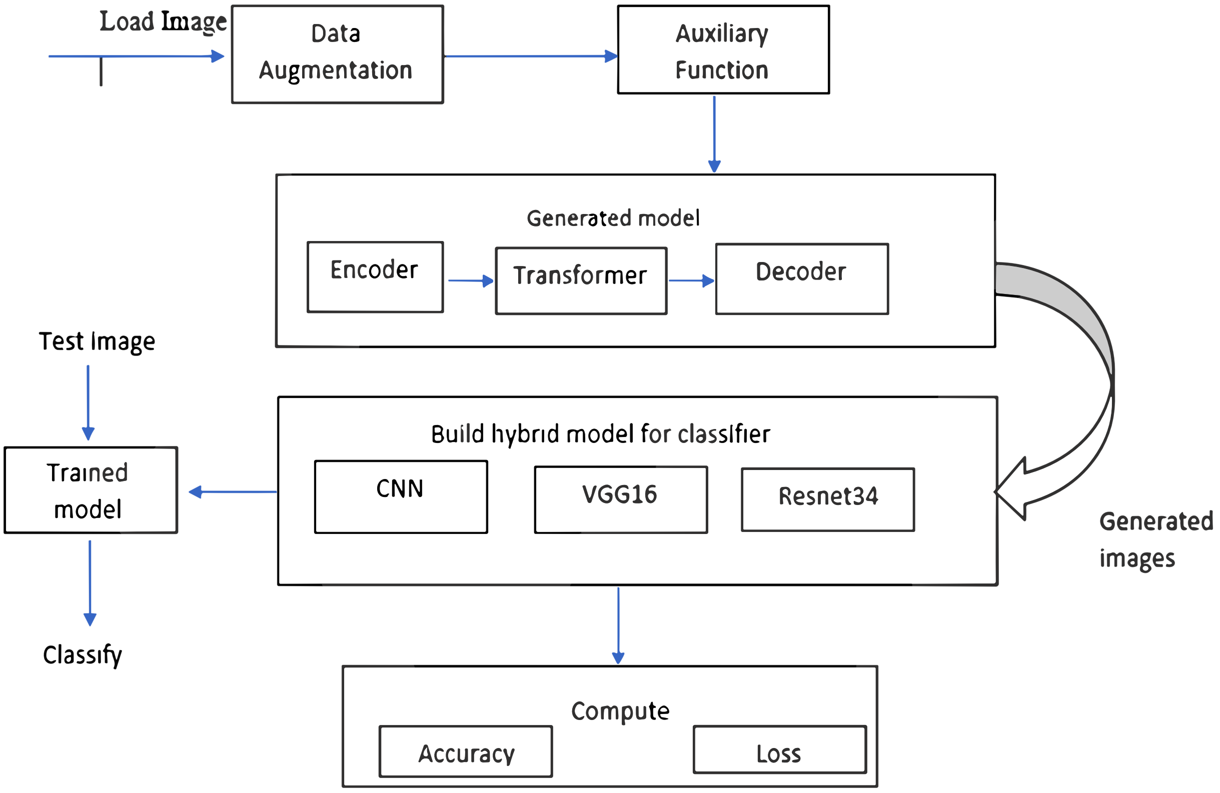


Fig. (2). The architecture of the proposed model.

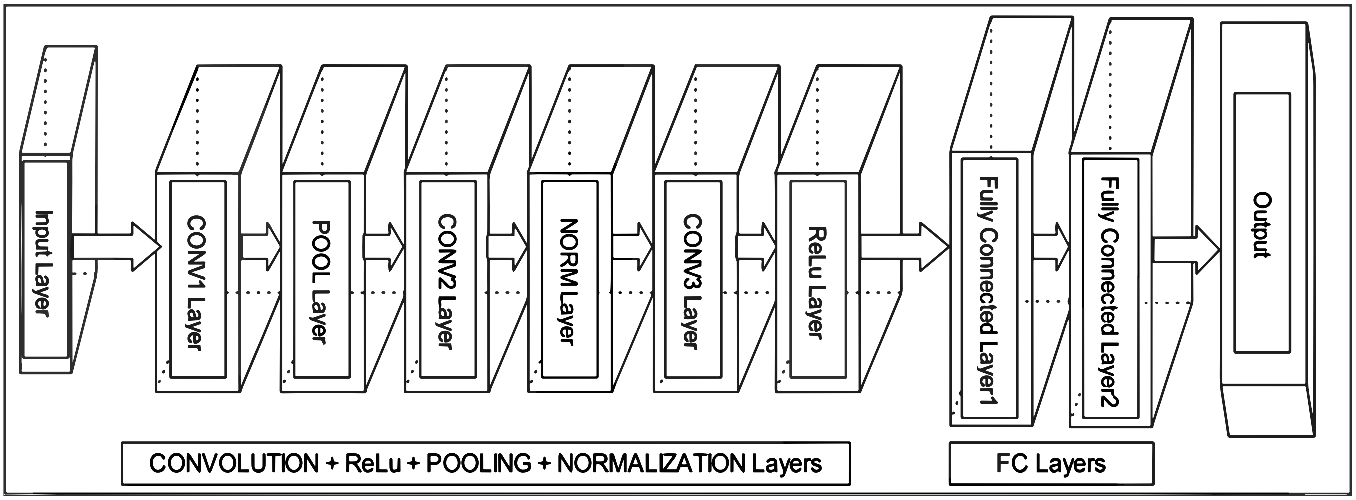


Fig. (3). CNN design.

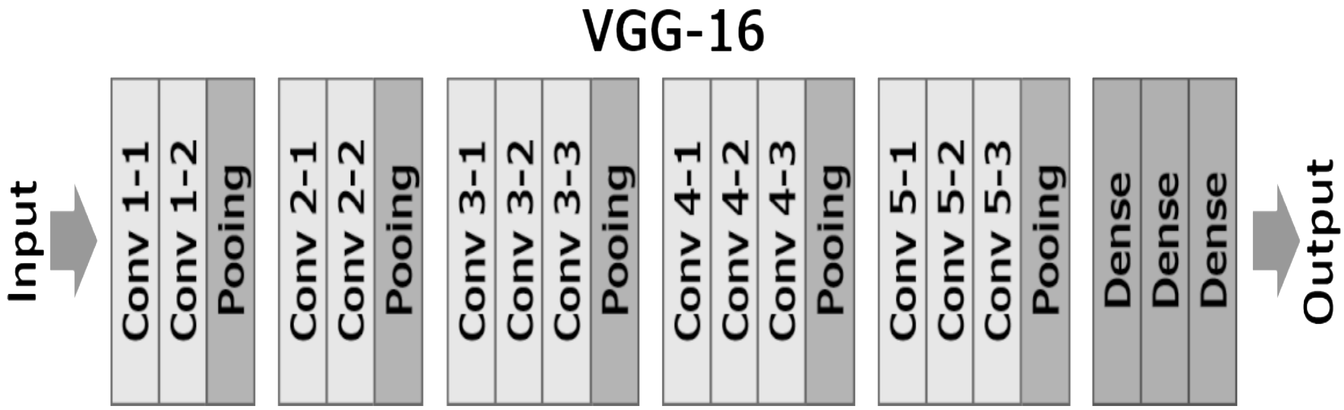


Fig. (4). VGG16 design.

Table 2. Description of cassava leaf dataset.

Label	Name	Number of Images
0	CBB	2576
1	CBSD	2576
2	CGM	2576
3	CMD	2576
4	Healthy	2576

4. RESULTS AND DISCUSSION

The dataset is accumulated from the Kaggle. The dataset has four types of unhealthy and one healthy leaf class [1]. The data label is shown in Table 2. 12880 JPG image files center center-cropped and converted into TFRecords.

4.1. Experimental Setup and Evaluation

Two units of experiments have been completed in this study. The CycleGAN model was trained for 30 epochs to generate false leaf pictures of cassava for each of the 5 classes in the first batch of experiments. After each epoch, the weights of the generator and discriminator models were adjusted to produce artificial images that were as close to genuine photographs as possible. In the second set of experiments, the combination of CNN, VGG16, and ResNet34, CNN+ResNet34, CNN+VGG16, and VGG16+ ResNet34 models have been trained on the combination of the original set and generated synthetic cassava leaf photographs. Overall performance metrics like validation accuracy, validation loss, and training loss are used to evaluate the performance. This work created two versions of the TFRecords, one version of the TFRecord resized the original 600x800 images, and another one that first center-cropped the images to 600x600 before resizing and then resizing them. Table 3 shows how the 600x800 original images were resized by the version of the TFRecords.

Table 3. Image resizing by the version of the TFRecords.

Resolution	Resized	Center Cropped	External Center Cropped
128x128	15 TFRecords	15 TFRecords	15 TFRecords
256x256	15 TFRecords	15 TFRecords	15 TFRecords
384x384	15 TFRecords	15 TFRecords	15 TFRecords
512x512	15 TFRecords	15 TFRecords - 50 TFRecords	15TFRecords - 50 TFRecords

Table **4** shows the added last dataset that is divided by classes, it might be useful if we want to do some kind of class sampling or train Generative Adversarial Networks.

Table 4. Image resizing by the version of the TFRecords.

Resolution	Center Cropped	External Center Cropped
512x512	15TFRecords - 50 TFRecords	15TFRecords - 50 TFRecords

The files were divided into 15 TFRecords files to make it easier to do K-Fold splits, they were stratified by the label. Evaluating a Machine Learning model may be quite complicated. Usually, we break up the records set into training and testing sets. We then compare the model performance based on an error metric to determine the accuracy of the version.

This method, however, isn't always very reliable as the accuracy acquired for one check set can be very exclusive from the accuracy obtained for a special test set. K-fold Cross-Validation (CV) affords a way to this problem by way of dividing the information into folds and making sure that each fold is used as a checking-out set sooner or later.

K-Fold CV is where a given statistics set is split right into a K range of sections/folds wherein every fold is used as a checking out set sooner or later. Table **5** shows how the files were divided to do K-Fold splits.

Table 5. K-Fold splits.

File	Sample
1	1427
2	1427
3	1427
4	1427
5	1427
6	1427
7	1427
8	1426
9	1426
10	1426

Fig. (**5**) shows the example of the label's distribution comparing the complete set to one of the TFRecord files.

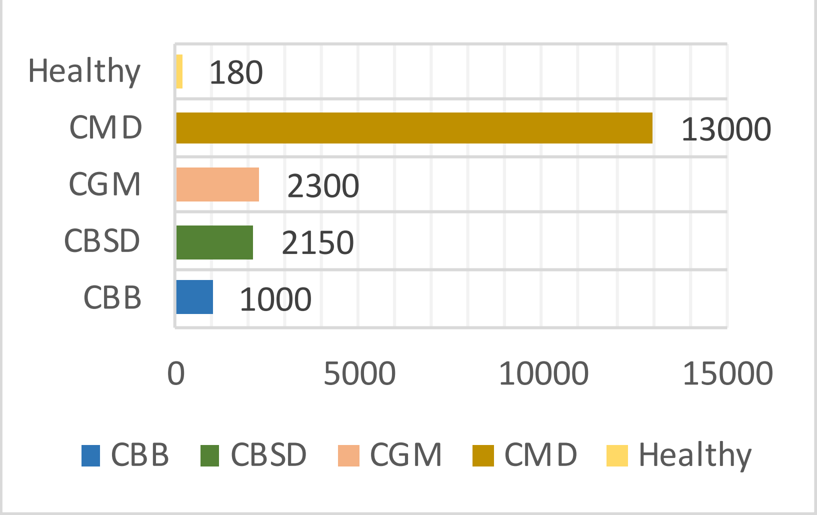
4.1.1. Augmentations

Data augmentation for GANs should be done very carefully, especially for tasks similar to style transfer, if we apply transformations that can change too much the style of the data (*e.g.*, brightness, contrast, saturation) it can cause the generator to not efficiently learn the base style, so in this case, we are using only spatial transformations like, flips, rotates and crops.

4.1.2. Auxiliary Function

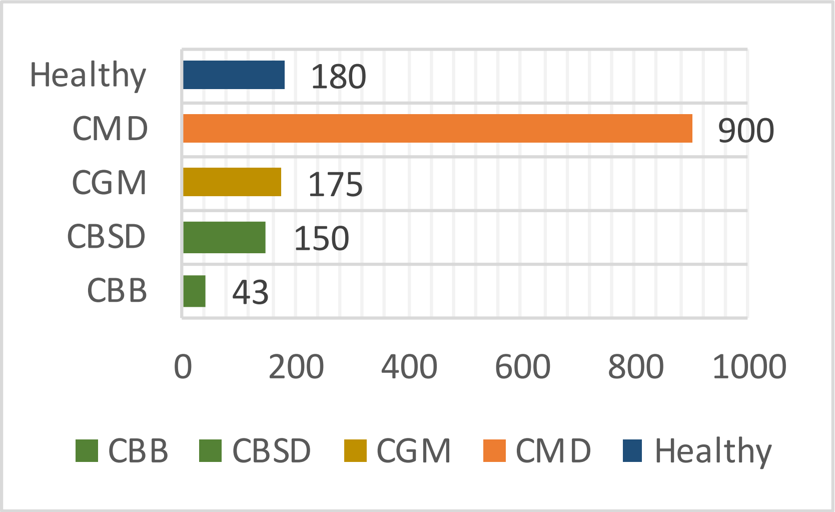
Here developing the blocks of the models. There are three blocks as shown below

- Encoder block: Apply convolutional filters while also reducing data resolution and increasing features.
- Decoder block: Apply convolutional filters while also increasing data resolution and decreasing features.
- Transformer block: Apply convolutional filters to find relevant data patterns and keep features constant.



(a) Complete setl abel distribution

Fig. 5 contd.....



(b) Labels distribution fore ach file

Fig. (5a, b). Complete set label distribution v/s labels distribution for each file.

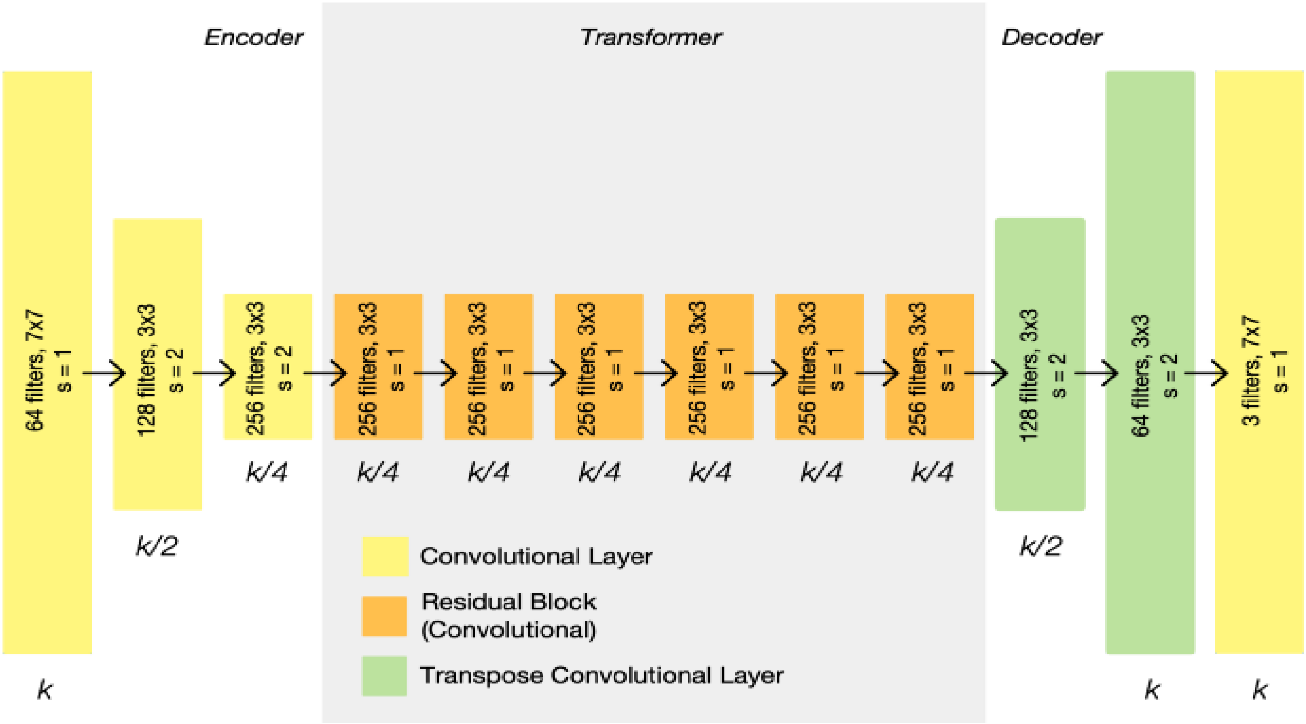


Fig. (6). CycleGAN model [2].

4.1.3. Generator Model

- The generator is responsible for generating images from a specific domain. CycleGAN architecture has two generators, in this context for example we can take one generator that will take Healthy images and generate CBB images, and the other generator will take CBB images and generate Healthy images.
- Below Fig. (6), is the architecture of the original CycleGAN generator, we have some changes to improve performance on this task.

Now Build the CycleGAN Model and Generator Models for respective Categories and save the models as cbb_generator.h5, cbsd_generator.h5, cgm_generator.h5, cmd_generator.h5, and healthy_generator.h5.

With the help of Generator Models, we can generate the images category-wise

The generator mapping functions are as follows:

$$A: U \rightarrow \mathrm{V} \tag{4}$$

$$F : V \rightarrow U \tag{5}$$

where X is the input image distribution and Y is the desired output distribution.

CycleGAN has two mapping functions A: U→V and F: V →U corresponding to data domains U and V. The education of A calls for a discriminator DV to classify the generated photograph A(u) from the real samples yi ∈ V. The mapping F and the corresponding discriminator DU, which discriminates the generated image F(y) from the actual samples xi ∈ U, are also educated concurrently. We expect right here that U and V are the sets of wholesome and arbitrary target disorder pics, respectively.

The input is passed into the encoder, it extracts characteristics from the picture utilizing convolutions and compressing the representation of the photograph but will increase the number of channels. Three convolutions are included in the encoder, each of which uses 1/4th of the real picture length to minimize the illustration. The encoder's output is transmitted through the transformer once the activation characteristic is applied. Depending on the size of the input, the transformer may have six or nine residual blocks. The transformer's output is then transmitted *via* the decoder, which utilizes a fraction strides convolution block to raise the representation's dimensions to the authentic length.

4.2. Performance Evaluation of CycleGAN

The performance of CycleGAN can be visually examined in Fig. (7). This shows the unique cassava leaf photographs from the dataset and artificial cassava leaf pics generated by using the Cycle GAN model.

After getting the generated Images from GAN techniques for different categories, we load those categories of images for classification purpose.

The combination of CNN, VGG16, and Resnet34 models are used for the classification of different categories of cassava leaf images. The dataset contains 12880 images. 11587 images were used for training and 1023 images for testing. Fig. (8) shows the inversion of the original image.

Fig. (9) presents the experimental outcomes *i.e* graphical plot of training accuracy, validation and training loss of the various proposed model.

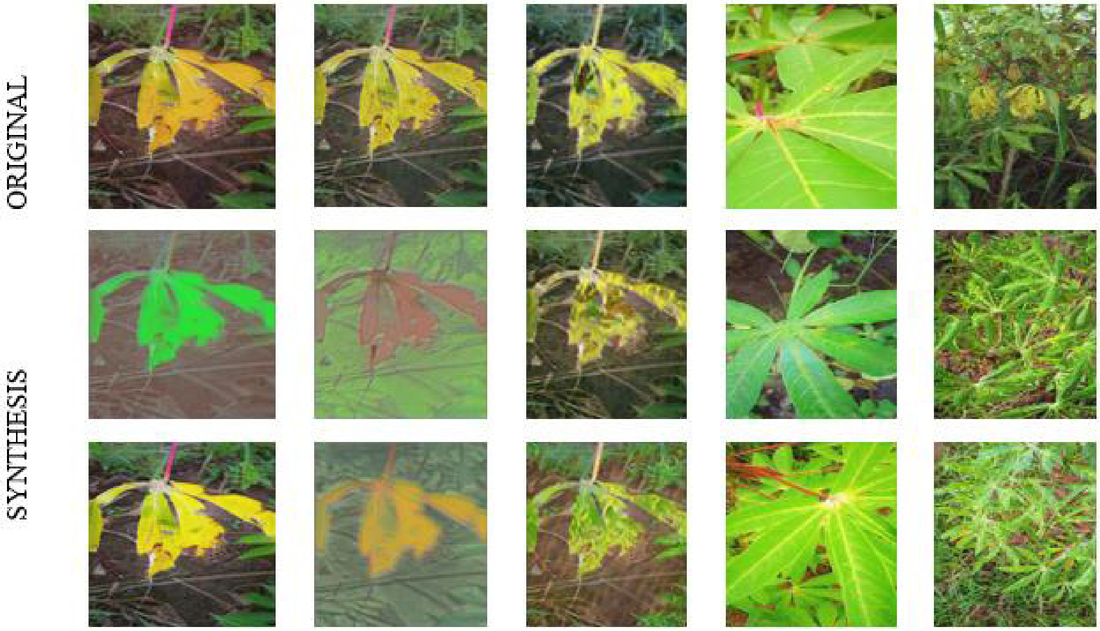


Fig. (7). Comparison of original image and synthesis image.

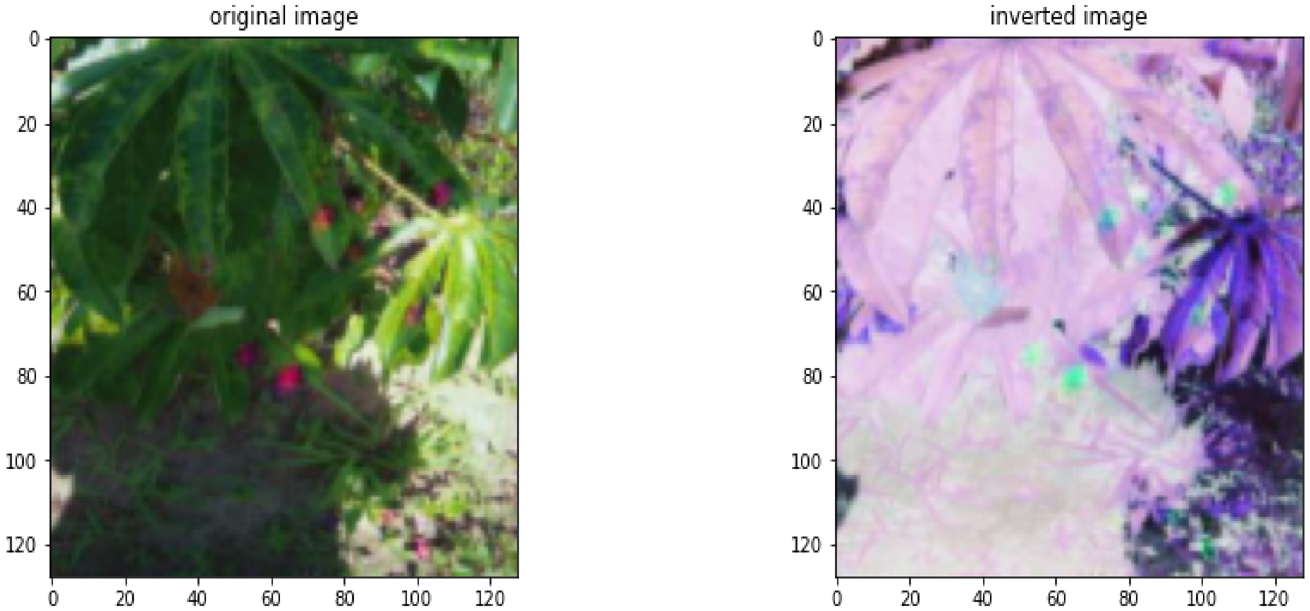
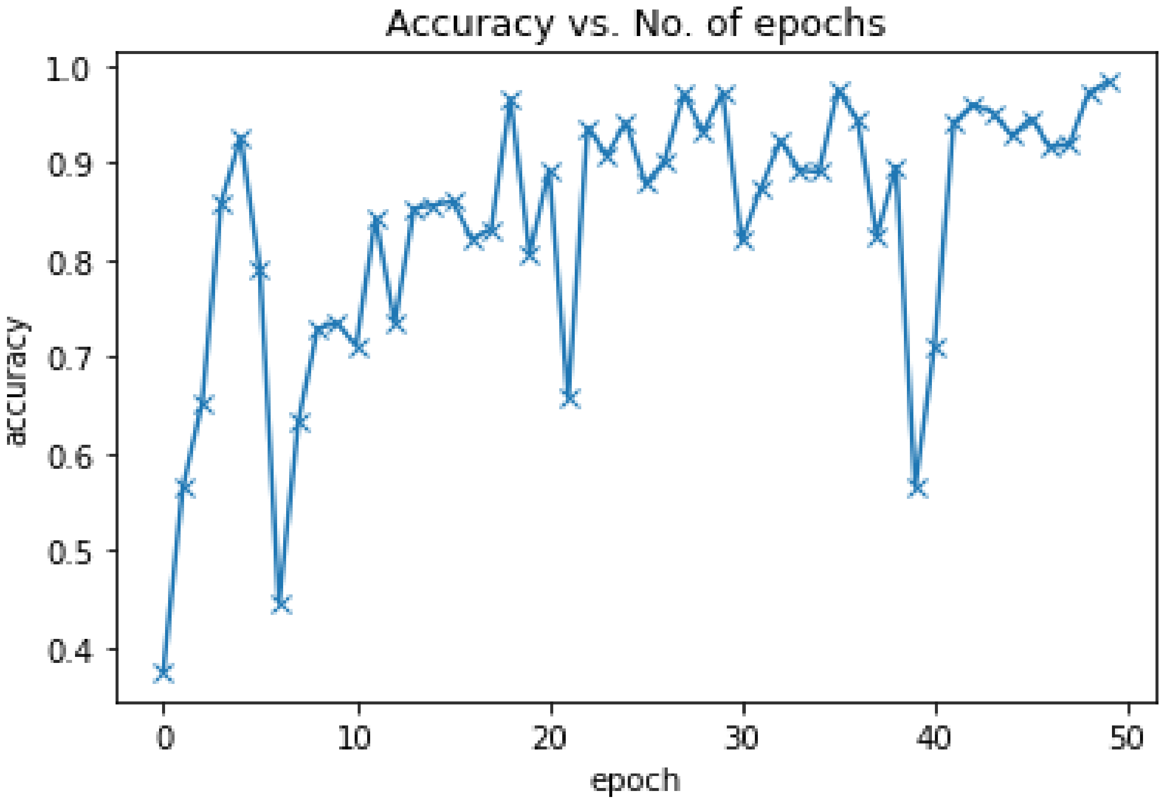
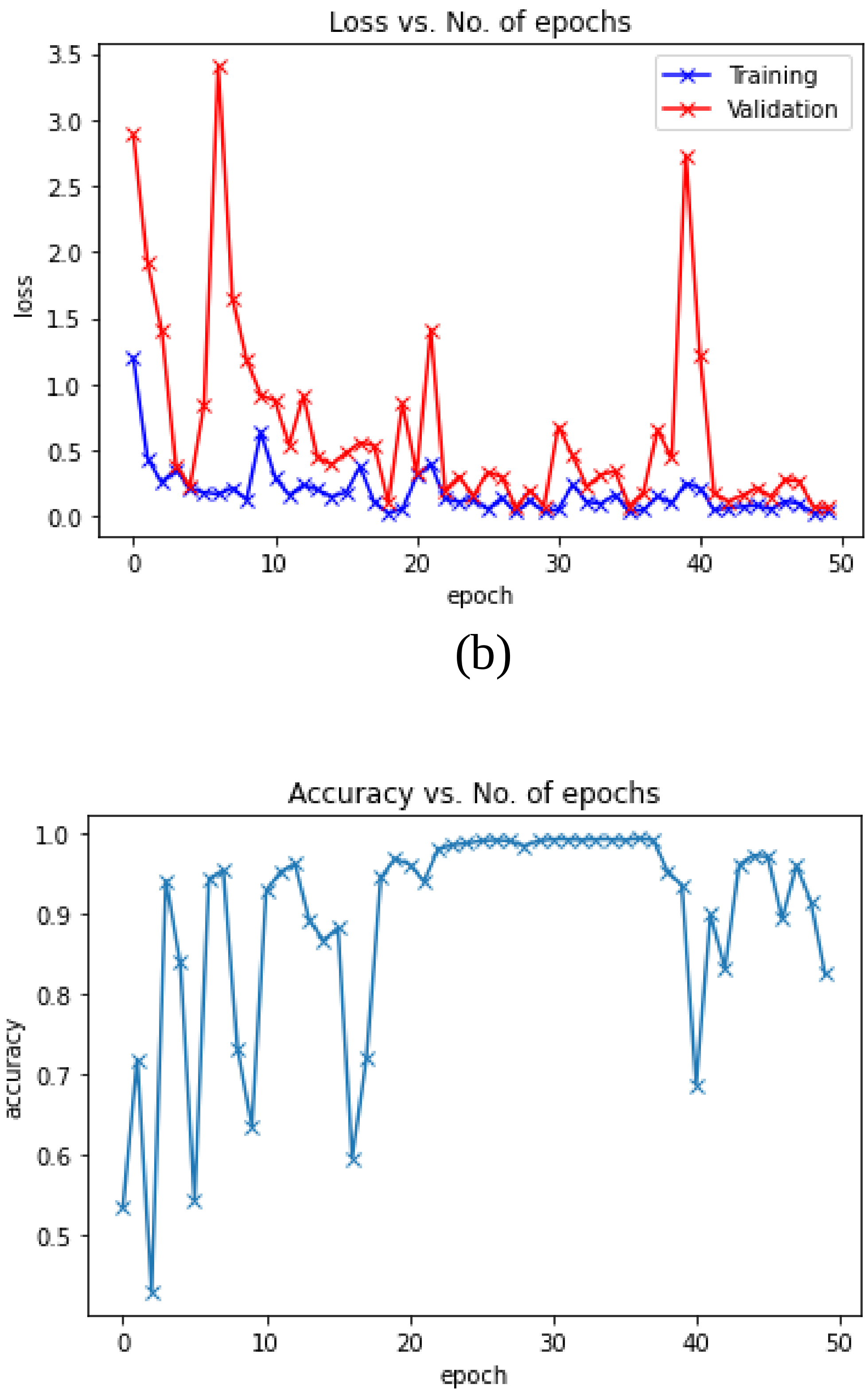


Fig. (8). Inversion of the original image.



(a)

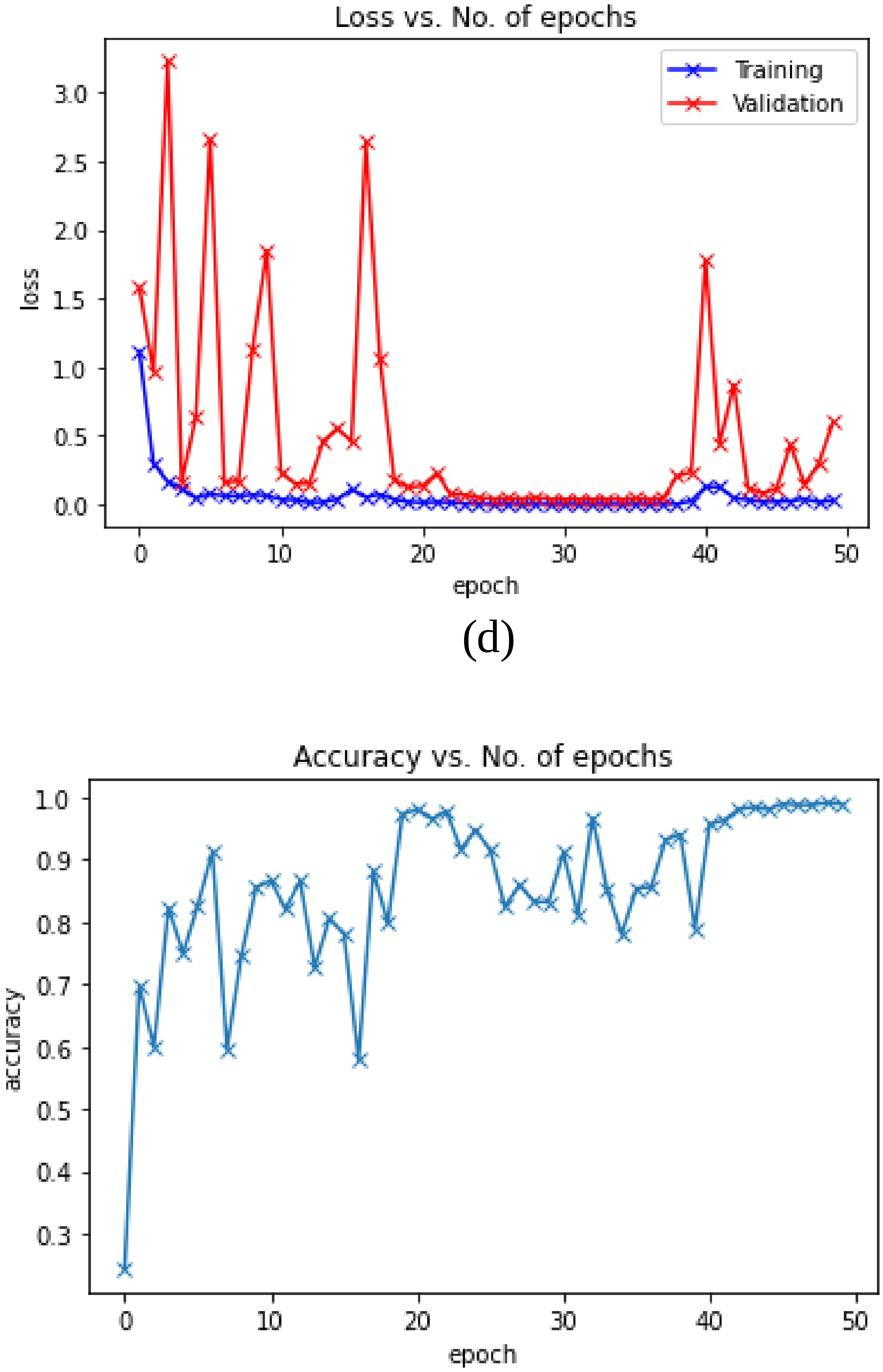
Fig. 9 contd.....



(b)

(c)

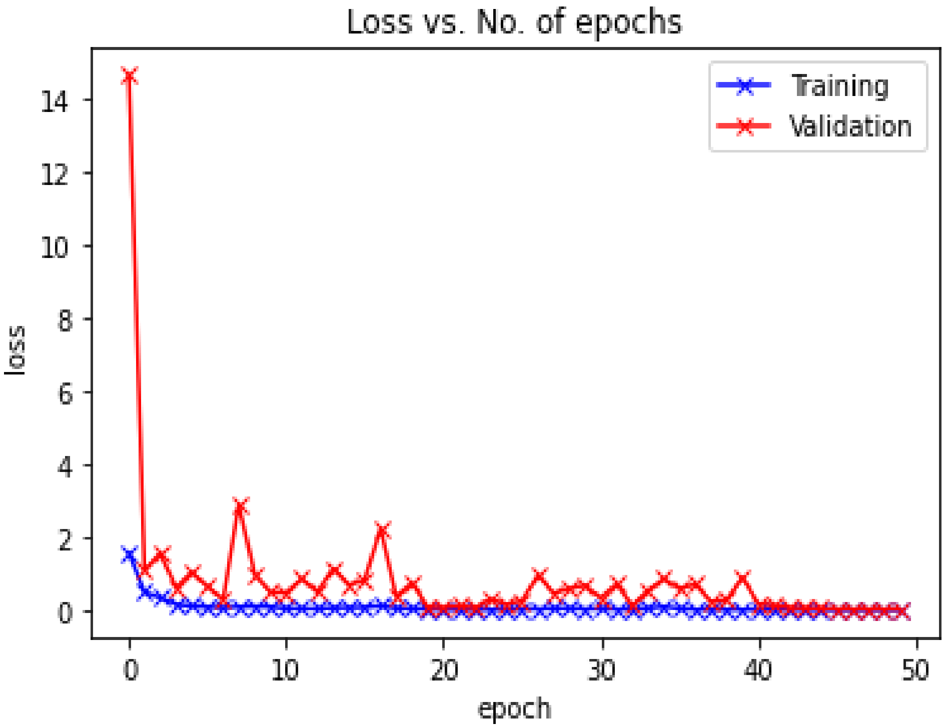
Fig. 9 contd.....



(d)

(e)

Fig. 9 contd.....



(f)

Fig. (9). Experimental Outcome of Proposed Models (**a**) CNN+VGG16 Training Accuracy, (**b**) CNN+VGG16 Training and Validation Loss, (**C**) CNN+ResNet34 Training Accuracy, and (**d**) CNN+ResNet34 Training and Validation Loss (**e**) VGG16+ResNte34 Training Accuracy (**f**) VGG16+ResNet34 Training and Validation Loss).

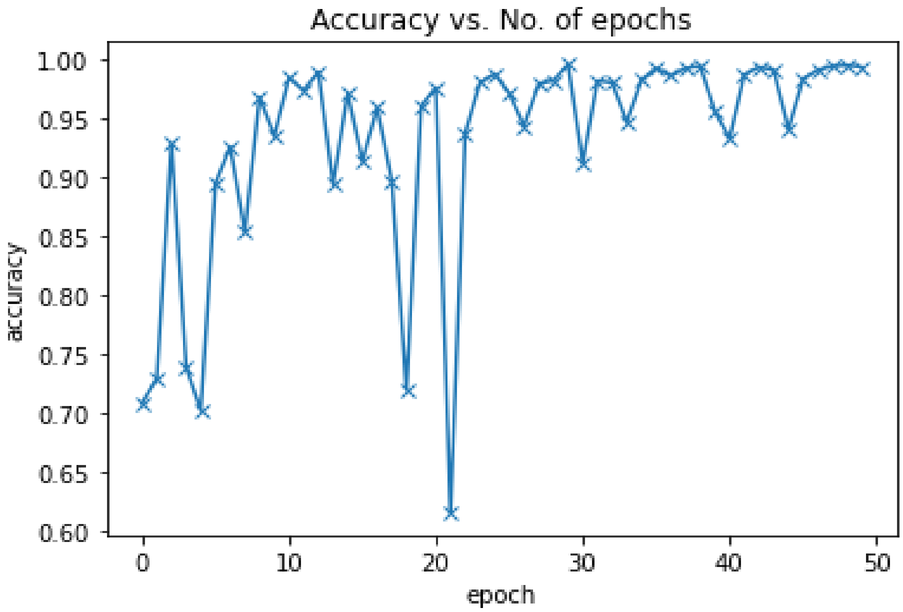


Fig. (10). Accuracy *vs* epoch.

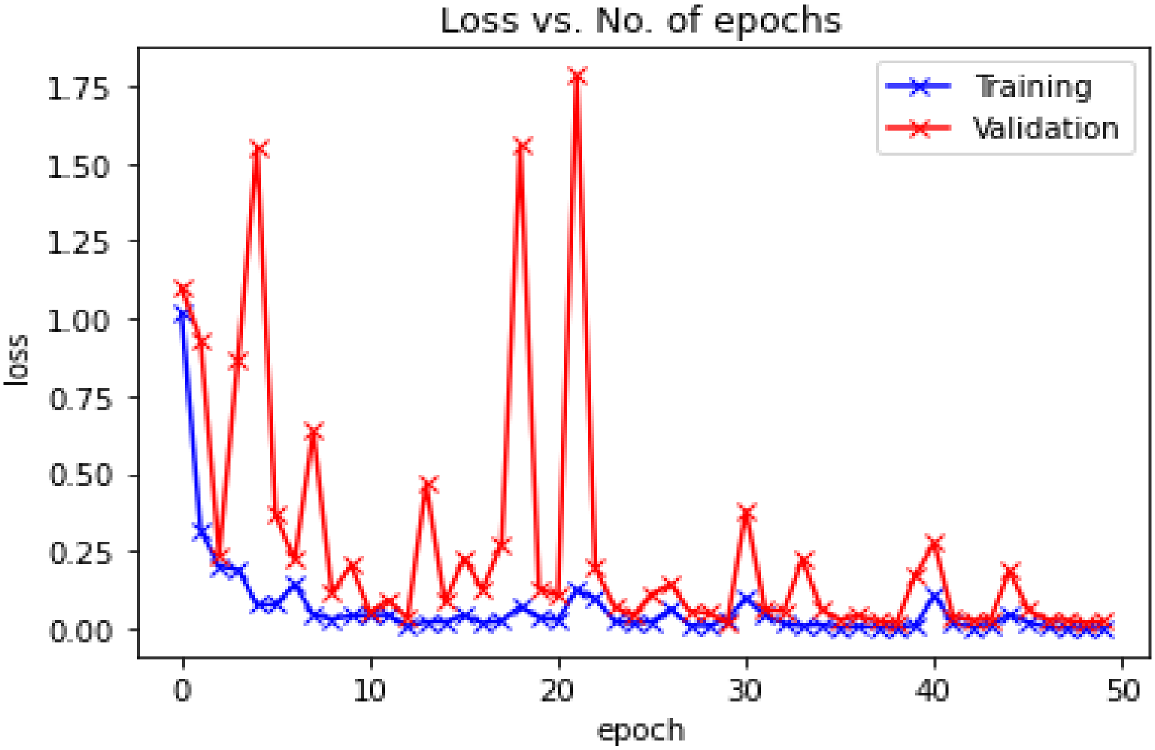


Fig. (11). Loss *vs* epoch.

Fig. (**10**) shows the Accuracy *vs*. Number of epochs. We trained the model for 50 epochs, and the model (CNN+VGG16+ResNet34) was able to achieve the best accuracy of 99.51% at epoch number 30.

Fig. (**11**) shows the graphical plot of the training loss and validation loss concerning several epochs of the hybrid model (CNN+VGG16+ResNet34).

Table **6** indicates the experimental results for diverse Cassava leaf disease training (zero to four) for the proposed hybrid version for the dataset. These experimental results display that the proposed version performs higher accuracy, precision, and f1-score than the present method.

Fig. (**12**) shows the confusion matrix for various Cassava leaf sickness lessons using the proposed hybrid version (CNN+VGG16+ResNet34). This matrix shows the results of actual *vs*. forecasted statistics. The accuracy of the healthy class is 99.64 percent, which is superior to the other classes.

The GUI is developed to predict the class of disease. It has two options, browse for the input image and classify the test image. The user has to upload the input image by clicking on browse for the input image. After uploading when the user clicks on classified image it will display the type of disease. The development of a GUI for disease classification allows users to easily upload and classify test images, eliminating the need for manual coding or complex command-line interfaces. This simplifies the process and reduces potential costs associated with technical expertise or training.

5. PERFORMANCE COMPARISON

Several studies have been proposed in the past for detecting cassava plant illness from cassava leaf snap photos. We compared our proposed approach to several recent studies that have been proposed using the cassava dataset.

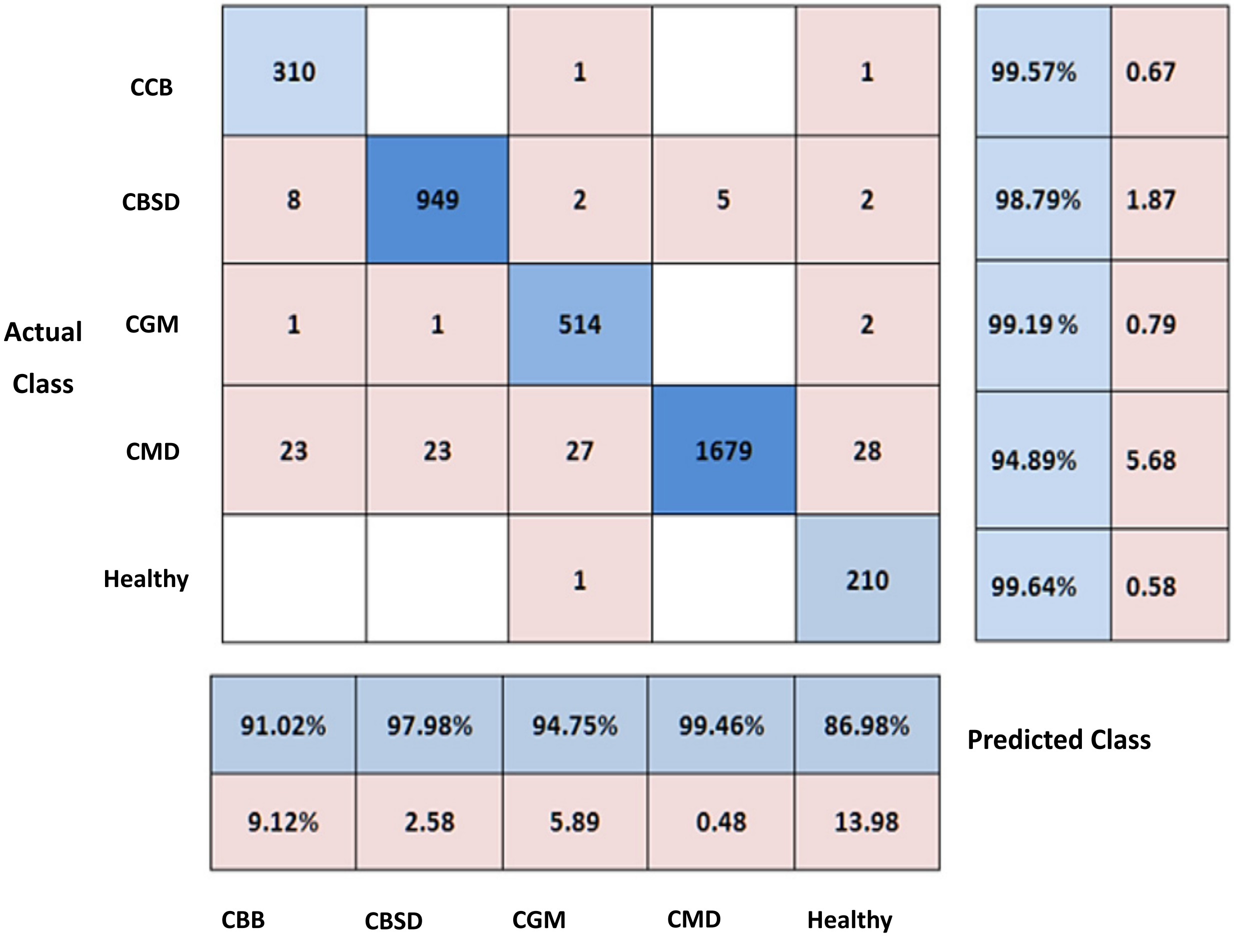
Table **7** examines performance evaluation using current approaches. The research presented in the table is particularly advanced for extraordinary illness type demands. The proposed hybrid model CNN+VGG16+ResNet34 was able to achieve much higher accuracy compared to existing hybrid models.

Table 6. Experimental results.

Class	Precision	Accuracy	Recall	F1-score
CBB	90.21	92.30	84.56	84.35
CBSD	97.98	95.25	93.65	85.71
CGM	94.32	95.64	72. 13	76.45

(Table 6) contd.....

Class	Precision	Accuracy	Recall	F1-score
CMD	99.12	98.60	97.23	85.32
Healthy	96.50	97.77	91.34	92.75



Actual Class	CCB	310		1		1	99.57%	0.67
	CBSD	8	949	2	5	2	98.79%	1.87
	CGM	1	1	514		2	99.19 %	0.79
	CMD	23	23	27	1679	28	94.89%	5.68
	Healthy			1		210	99.64%	0.58
		91.02%	97.98%	94.75%	99.46%	86.98%	Predicted Class	
		9.12%	2.58	5.89	0.48	13.98		
		CBB	CBSD	CGM	CMD	Healthy		

Fig. (12). Confusion matrix.

Table 7. Performance comparison for Cassava leaf disease detection.

Author(s)	Methods	Accuracy %
Ayu *et al*. [10]	MobileNet	65.6
Ozichi *et al*. [12]	SVM	83.9
Amanda Ramcharan *et al*. [8]	Inception V3	93
Sambasivam *et al*. [9]	CNN	85.3
Agbotiname Lucky *et al*. [25]	ECNN	95.65
Proposed	CNN+VGG16+ResNet34 CNN+VGG16 CNN+ResNet34 VGG16+ResNet34	**99.51** 96.90 98.57 99.03

CONCLUSION

Plant disease is a serious problem in agriculture, therefore researchers are looking for a way for its automatic early detection, which we addressed through modern AI and ML techniques applied to computer vision and image recognition. In this study, we created a hybrid model for predicting illness class, and we have introduced another deep learning technique for detecting cassava plant disease from cassava leaf photos. For data augmentation, synthetic images were generated using Cycle Generative Adversarial Networks. The present standard CNN models use a larger number of features and a more complex computational procedure, which increases the computational cost. To address this issue, we combined the traditional CNN approach with VGG16 and ResNet34. The proposed approach aims to reduce expenses related to labor, data collection, training, and technical expertise in leaf disease detection using GANs. The proposed model employs a depth-wise layer separation and implements K-Fold Cross-Validation to reduce feature count and computational overhead. The proposed hybrid model is compared to the existing standard approach *i.e*. enhanced CNN. An experiment was carried out on a leaf dataset. The analysis demonstrated that the proposed hybrid model (*i.e*., CNN+VGG16+ ResNet34) outperforms the existing models in terms of accuracy.

In the future, we plan to improve on the current results in a variety of ways: (a) The dataset size and the number of disease classes can be increased. (b) Additional performance evaluation parameters can be added to ensure real-time computation.

LIST OF ABBREVIATIONS

GAN	=	Generative Adversarial Networks
DL	=	Deep Learning
ML	=	Machine Learning
CNN	=	Convolutional Neural Network

CONSENT FOR PUBLICATION

Not applicable.

AVAILABILITY OF DATA AND MATERIALS

The data and supportive information are available within the article.

FUNDING

None.

CONFLICT OF INTEREST

The authors declare no conflict of interest financial or otherwise.

ACKNOWLEDGEMENTS

Declared none.

REFERENCES

[1] Mathulaprangsan GO, Lanthong K. Cassava leaf disease recognition using convolutional neural networks. Proceedings of the 2021 9th International Conference on Orange Technology (ICOT),. IEEE, Tainan, Taiwan, 16–17 December 2021, pp. 1-5.

[2] Ahila Priyadharshini R, Arivazhagan S, Arun M, Mirnalini A. Maize leaf disease classification using deep convolutional neural networks. Neural Comput Appl 2019; 31(12): 8887-95. http://dx.doi.org/10.1007/s00521-019-04228-3

[3] Ali AA, Chramcov B, Jasek R, Katta R, Krayem S. Classification of plant diseases using convolutional neural networks. Artificial Intelligence in Intelligent Systems. Switzerland: Springer 2021; pp. 268-75. http://dx.doi.org/10.1007/978-3-030-77445-5_24

[4] Oyewola DO, Dada EG, Misra S, Damaševičius R. Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing. PeerJ Comput Sci 2021; 7: e352. http://dx.doi.org/10.7717/peerj-cs.352 PMID: 33817002

[5] Ravi V, Acharya V, Pham TD. Attention deep learning-based large-scale learning classifier for cassava leaf disease classification. Expert Syst 2022; 39(2): e12862. http://dx.doi.org/10.1111/exsy.12862

[6] Thai HT, Tran-Van NY, Le KH. Artificial cognition for early leaf disease detection using vision transformers. Proceedings of the 2021 International Conference on Advanced Technologies for Communications (ATC). , IEEE, Ho Chi Minh City, Vietnam, 14–16 October 2021, pp. 33-38. http://dx.doi.org/10.1109/ATC52653.2021.9598303

[7] Methil A, Agrawal H, Kaushik V. One-vs-all methodology based cassava leaf disease detection. Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT),. IEEE, Kharagpur, India, 6–8 July 2021, pp. 1-7. http://dx.doi.org/10.1109/ICCCNT51525.2021.9579920

[8] Uarrota VG, Moresco R, Coelho B, *et al.* Metabolomics combined with chemometric tools (PCA, HCA, PLS-DA and SVM) for screening cassava *(Manihot esculenta* Crantz) roots during postharvest physiological deterioration. Food Chem 2014; 161: 67-78. http://dx.doi.org/10.1016/j.foodchem.2014.03.110 PMID: 24837923

[9] Chen CC, Ba JY, Li TJ, Chan CCK, Wang KC, Liu Z. EfficientNet: A low-bandwidth IoT image sensor framework for cassava leaf disease classification. Sens Mater 2021; 33(11): 4031. http://dx.doi.org/10.18494/SAM.2021.3526

[10] Sambasivam G, Opiyo GD. A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. Egypt Info J 2021; 22(1): 27-34. http://dx.doi.org/10.1016/j.eij.2020.02.007

[11] Singh AK, Chaurasia B. Plant leaf disease detection using Convolutional neural network and random forest classifier. Int J Innov Eng 2021; 6(10): 204. http://dx.doi.org/10.46335/IJIES.2021.6.10.42

[12] Bose A, Ghosh D, Banerjee A, Saha D, Ganguly P, Chakrabarti S. Capsnet-VGG16 architecture for cassava plant disease detection. Proceedings of the International Conference on Computational Intelligence, Data Science and Cloud Computing (IEM-ICDC). Kolkata, India, 25–27 September 2020, pp. 62-207. http://dx.doi.org/10.1007/978-981-33-4968-1_17

[13] Zhang K, Wu Q, Chen Y. Detecting soybean leaf disease from synthetic image using multi-feature fusion faster R-CNN. Comput Electron Agric 2021; 183: 106064. http://dx.doi.org/10.1016/j.compag.2021.106064

[14] Mohameth F, Bingcai C, Sada KA. Plant disease detection with deep learning and feature extraction using plant village. J Comput Commun 2020; 8(6): 10-22. http://dx.doi.org/10.4236/jcc.2020.86002

[15] Jiang P, Chen Y, Liu B, He D, Liang C. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access 2019; 7: 59069-80.

http://dx.doi.org/10.1109/ACCESS.2019.2914929

[16] R K, M H, Anand S, Mathikshara P, Johnson A, R M. Attention embedded residual CNN for disease detection in tomato leaves. Appl Soft Comput 2020; 86: 105933.
http://dx.doi.org/10.1016/j.asoc.2019.105933

[17] Picon A, Seitz M, Alvarez-Gila A, Mohnke P, Ortiz-Barredo A, Echazarra J. Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. Comput Electron Agric 2019; 167: 105093.
http://dx.doi.org/10.1016/j.compag.2019.105093

[18] Gayathri Devi T, Neelamegam P. Image processing based rice plant leaves diseases in Thanjavur, Tamilnadu. Cluster Comput 2019; 22(S6): 13415-28.
http://dx.doi.org/10.1007/s10586-018-1949-x

[19] Liang Q, Xiang S, Hu Y, Coppola G, Zhang D, Sun W. PD 2 SE-net: Computer-assisted plant disease diagnosis and severity estimation network. Comput Electron Agric 2019; 157: 518-29.

[20] Tran TT, Choi JW, Le TT, Kim JW. A comparative study of deep CNN in forecasting and classifying the macronutrient deficiencies on development of tomato plant. Appl Sci 2019; 9(8): 1601.
http://dx.doi.org/10.3390/app9081601

[21] Surya R, Gautama E. Cassava leaf disease detection using convolutional neural networks. 6th International Conference on Science in Information Technology (ICSITech),. IEEE, Palu, Indonesia, 21-22 October 2020, pp. 97-102.
http://dx.doi.org/10.1109/ICSITech49800.2020.9392051

[22] Ramcharan A, Baranowski K, McCloskey P, Ahmed B, Legg J, Hughes DP. Deep learning for image-based cassava disease detection. Front Plant Sci 2017; 27: 1852.
http://dx.doi.org/10.3389/fpls.2017.01852 PMID: 29163582

[23] Sambasivam G, Opiyo GD. A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks Egypt Info J 2020; 22(1): 27-34.
http://dx.doi.org/10.1016/j.eij.2020.02.007

[24] Ayu HR, Surtono A, Apriyanto DK. Deep learning for detection cassava leaf disease. J Phys Conf Ser 2021; 1751(1): 012072.
http://dx.doi.org/10.1088/1742-6596/1751/1/012072

[25] Lilhore UK, Imoize AL, Lee CC, *et al.* Enhanced convolutional neural network model for cassava leaf disease identification and classification. Mathematics 2022; 10(4): 580.
http://dx.doi.org/10.3390/math10040580