Egyptian Informatics Journal 22 (2021) 27–34

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks

G. Sambasivam *, Geoffrey Duncan Opiyo

Faculty of Information and Communication Technology, ISBAT University, Kampala, Uganda

ARTICLE INFO

Article history:
Received 20 September 2019
Revised 11 February 2020
Accepted 20 February 2020
Available online 9 March 2020

Keywords:
Agriculture
Cassava mosaic detection
Rectifier Linear Unit
Synthetic minority over-sampling technique
Stochastic gradient descent

ABSTRACT

This work is inspired by Kaggle competition which was part of the Fine-Grained Visual Categorization workshop at CVPR 2019 (Conference on Computer Vision and Pattern Recognition) we participated in. It aimed at detecting cassava diseases using 5 fine-grained cassava leaf disease categories with 10,000, labeled images collected during a regular survey in Uganda. Traditionally, this detection is done mostly through physical inspection and supervision of cassava plants in the garden by farmers or agricultural extension workers from NAADS (National Agricultural Advisory Services) and then reported to NARO (National Agricultural Advisory Services) for further analysis. However, this can be tiresome, capital intensive, and lacks the ability to detect cassava infection timely to help farmers apply preventive techniques to the non-infected cassava plants in order to improve on yields which subsequently increases African food basket leading to food security which fights famine. Using the dataset provided to train CNNs (Convolutional Neural Networks) to achieve high accuracy was very challenging due to two reasons: the dataset was small in size and has high-class imbalance being heavily biased towards CMD (Cassava Mosaic Disease) and CBB (Cassava Brown Streak Virus Disease) classes. Class imbalance is problematic in machine learning and exists in many domains. Note that, not all world data is balanced, in fact, most of the time you will not be extremely lucky to get a perfectly balanced real-world dataset, in recent years, a lot of research has been done for two-class problems such as fraudulent credit card and tumor detection among others. Interestingly, class imbalance in multi-class image datasets has received little attention. This paper, therefore, focused on techniques to achieve an accuracy score of over 93% with class weight, SMOTE (Synthetic Minority Over-sampling Technique) and focal loss with deep convolutional neural networks from scratch. The goal was to counter high-class imbalance so that the model can accurately predict underrepresented classes.

1. Introduction

Cassava (*Manihot esculenta* Crantz) is one of the most common staple food crops grown in sub-Saharan Africa. Common parts of the plant that can be eaten are leaves and starchy roots; the starchy roots are by far the most commonly consumed because they are a valuable source of energy and can be eaten raw, roasted on a charcoal stove, boiled or processed in different ways for human consumption [1,7]. The leaves and tender shoots are a rich source of proteins and vitamins and are consumed as a vegetable in many regions [2]. Farmers across Africa are growing cassava in small, medium to large scale under a wide range of environmental and climatic conditions contributing to food security and industrial crop, but the major challenge is that cassava plants are vulnerable to a broad range of diseases as well as less known viral strains. In Uganda, the most common are CMD which shows symptoms of yellowing and wrinkled leaves and CBB which leads to root rot are among the serious threats to Sub-Saharan Africa's food security. Cassava Mosaic infection was first reported in Tanzania towards the end of the 19th century [18] and from that time, the epidemic has spread throughout Sub-Saharan Africa resulting in great economic loss and devastating famine because they are the major constraints to cassava production [28–31]. This means we need better-automated approaches that can assist farmers in early detection and prevention of cassava diseases because conventional plant dis-

* Corresponding author.
E-mail address: gsambu@gmail.com (G. Sambasivam).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

https://doi.org/10.1016/j.eij.2020.02.007
1110-8665/

ease diagnosis by human experts is tiresome, capital intensive, and lacks the ability to detect cassava infection timely [34].

Our model performance showed promising results and to the best of our knowledge, there is still no experimentation done for classification of cassava mosaic and other cassava disease detection by training CNNs from scratch using an imbalanced dataset. This work presents our techniques used to achieve an accuracy score of over 93% with a very limited image dataset of 5 fine-grained cassava leaf disease categories having 10000, labeled images. Additionally, these images were highly imbalanced, being heavily biased towards classes – CMD and CBSD.

2. Literature review

2.1. Disease categories

Four of the most common cassava diseases are shown in Fig. 1 below. Each has unique signs and symptoms that appear on the leaf, these can be used to differentiate and categorize the infections both visually by human eyes and automatically by deep learning algorithms [3–5,21].

2.2. CGM (cassava green mite)

It causes white spots on the leaves. It starts with small spots which then enlarges to cover the whole leaf surface leading to loss of chlorophyll hence affecting photosynthesis. Severe CGM also causes mottling symptoms that can be easily confused with cassava mosaic, the leaves affected dry out, shrink and break away from the plant.

2.3. CBSD

Is carried by a vector called whiteflies. Its symptoms include characteristic yellow of the vein which sometimes enlarges and forms visually large yellow patching [35]. CBSD also shows symptoms of dark-brown necrotic areas on the tuber root with a reduction in root size.

2.4. CMD

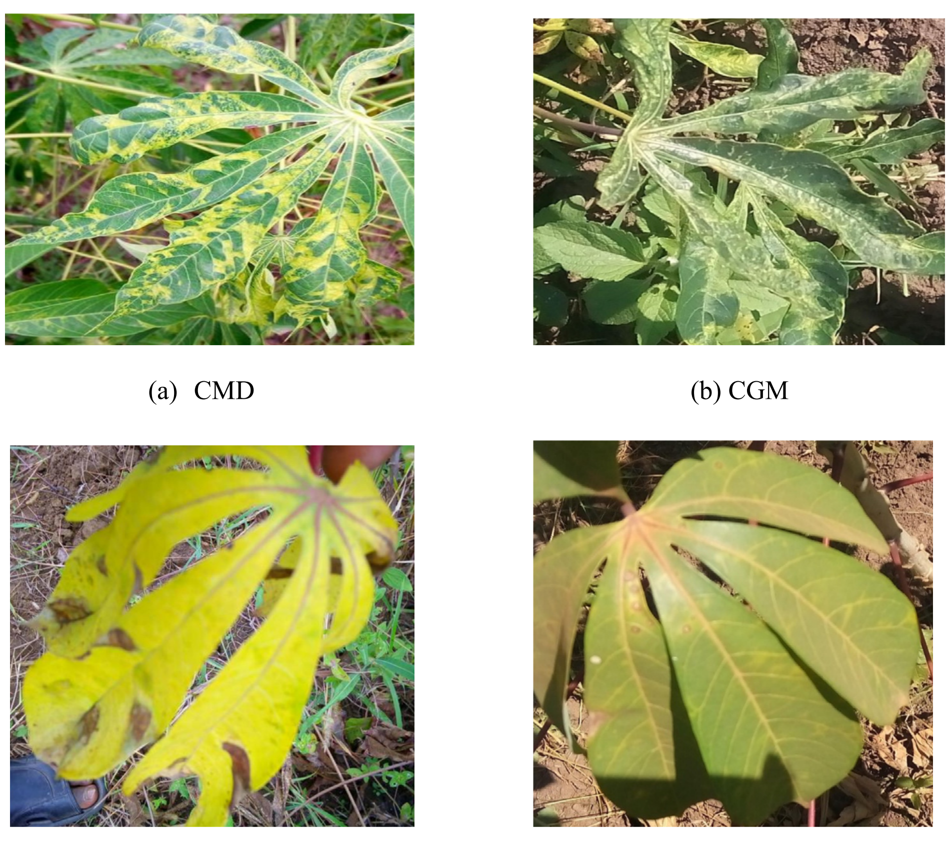
Has many foliar symptoms of mottling, mosaic rust, twisted leaves and a general reduction in sizes of leaves as well as the affected plants. Leaves always have patches of green mixed with the different coloring of yellow and white patches [36]. The patches reduce the surface area for photosynthesis resulting in stunted growth and low yield.

2.5. CBB (cassava bacterial blight)

This is a bacterial infection attributed by moisture. So, cassava plants in moist areas are the ones most affected. The symptoms exhibited are black leaf spots and blights. The affected leaves dry prematurely and shed-off as a result of wilting.

2.6. Related works

Deep learning for plant disease diagnosis in the early works [8–10,19] all used leaf images as a data source but didn't specify whether the datasets were balanced or imbalanced. [19] used spectroscopy aimed at studying how different materials interact with light in respect to wavelengths that will be absorbed or reflected, the study argued that most of the early works on used leaf images



(a) CMD (b) CGM

(c) CBB (d) CBSD

Fig. 1. Cassava infections as visually viewed by human eyes from the dataset.

as the key data input but by the time a symptom is shown, it means the disease is already at an advanced stage and from a practical point of view nothing can be done to save the affected plants hence necessity for alternative means such as spectroscopy to detect and curb infection at early stages. [8], introduced a practical and applicable solution for detecting the class and location of diseases in tomato plants. The goal was to find more suitable deep-learning architecture processing units rather than the method of collecting physical samples (leaves, plants) and analyzing them in the laboratory as done by earlier works. In [9], researchers Amanda Ramcharan, Peter McCloskey, Kelsee Baranowski, and David Hughes used Inception v3 transfer learning on a dataset of cassava disease images taken from the field in Tanzania to train a deep convolutional neural network to identify three diseases and two damages from pests, they have proven that transfer learning is so good a tool in automated disease detection. The model was deployed on mobile devices to detect infections in cassava plants in real-time via a TensorFlow application. In [10], several model architectures such as AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat and VGG (Solid State Drive) were trained on an open database containing 87,848 images, having 25 different plants in a set of 58 distinct classes of plant disease combinations, including healthy plant., with the best performance reaching a 99.53% success rate. Our experiment builds on these previous works as discussed above but using a small dataset with high class imbalanced as shown in Fig. 4.

3. Proposed methods

The proposed algorithm for this paper was Convolutional Neural Networks to build a low-cost method to detect cassava infections through deep-learning [11] with the implementation steps in the sequence of dataset acquisition, labeling, training the model, testing/model evaluation using k-fold cross-validation, where k = 3 to achieve the desired accuracy.

The model building steps taken were:

a) Environment preparation, loading and pre-processing Data – 35% time
b) Defining Model architecture – 10% time
c) Training the model – 50% time
d) Estimation of performance – 5% time

The problem of this dataset was the size, very small which could lead the model to suffer from overfitting problems. Additionally, the classes were highly imbalanced being heavily biased towards CMD, CBSD classes and images were having poor resolution and low contrast. Deep learning techniques proposed in this study for addressing class imbalance to counter bias were algorithm method using class weight [12], focal loss in Lin et al. [13] as a loss function for addressing extreme class imbalance and hybrid-method through resampling minority class using SMOTE (Synthetic Minority Oversampling Technique) [14,15] discussed in detail in Section 4.3.

4. Experimental setup

4.1. Dataset

Images used in the experiment were adopted from Kaggle [17,16]. This dataset consisted of 5 fine-grained cassava leaf disease categories with 10,000 labeled images collected during a regular survey in Uganda, mostly crowdsourced from farmers taking images of their gardens, and annotated by experts at the National Crops Resources Research Institute in collaboration with Artificial Intelligence lab in Makerere University, Kampala. In order to be able to train the model successfully, we needed to find ways to preprocess the input images to improve contrast and secondly, we needed to counter class label skew (imbalanced in the dataset) as illustrated in Fig. 4.

4.2. Model architecture

The architecture of the model used was composed of 3 convolutional layers and head of 4 fully connected layers. The first layer having 32 5×5 kernels to learn larger features, batch normalization and max-pooling of 3×3 pooled size. Second and third layers each consisted of two sets of convolutional layers each having 64 3×3 and 128 3×3 feature detectors, batch normalization, and max-pooling layers respectively. The layers were organized this way in order to allow the network to learn richer features by stacking together two sets of convolutions and batch normalization layer before max-pooling. The model architecture is shown in Fig. 2. The purpose of the max-pooling layer is to apply volume spatial dimensions reduction to the input images. The head of the network consisted of 4 fully connected layers with 512 neurons in the first layer, 1024 neurons in the second and third layers and lastly 256 neurons in the fourth layer and a neuron per every category in the output layer corresponding to five different classes after parameters tuning and optimization with grid search. Dropout was used in the fully connected layers as regularizers [22,23] to reduce generalization error and over-fitting problems by encouraging the neural network to learn sparse features of raw observations which always yields good performance by empowering model's ability to generalize to new data. In general, the convolutional layers extract key features from the images and the fully connected layers focus on using the extracted features to classify images of cassava leaves into five different categories. Input image attributes takes an order 3 tensor, e.g., an image with H rows, W columns, and 3 channels (R, G, B color channels) on the input layer and a neuron per every category in the output layer corresponding to five different classes [cmd, cbsd (Bacteria Blight of Cassava), cgm (Green Mite infection), healthy and cbb]. The activation functions used in convolutional and hidden layers were ReLU (Rectifier Linear Unit) and the output layer was softmax (for a multi-class case) not sigmoid (for a binary case) function.

4.3. Environment preparation, data-preprocessing and model training

Turning the tide of this experiment started by installing 64-bits Ubuntu 16.04.6 LTS (Xenial Xerus) on a laptop having specifications of 8 GB RAM (Random Access Memory), 200 GB (Gigabyte) SSD (Solid State Drive) hard disk and Intel® Core™ i7-7500U CPU (Central Processing Unit) processor followed by Anaconda IDE (Integrated Development Environment) installation and all the necessary libraries including Python 3.7, Scikit-learn, Numpy, Tensorflow-2.0.0 none GPU (Graphics Processing Unit) version, Matplotlib and OpenCV (Open Computer Vision).

Raw color images of both healthy and unhealthy cassava leaves in the Joint Photographic Exper Group (JPEG) file was split into 5 directories representing each class (multi-class) other than splitting the images into healthy and unhealthy directories only (binary-class). This way, it would enable the classification of four different cassava disease categories of the high-impact yet challenging problems affecting agriculture. Unfortunately, there were a number of challenges with the dataset: The first one was, dataset being small in size, the second challenge being images having low resolution and poor contrast and the last most challenging was class label skew in that, the top class has 44.45% while the least represented class has 3.16%, revealing an order of magnitude dif-

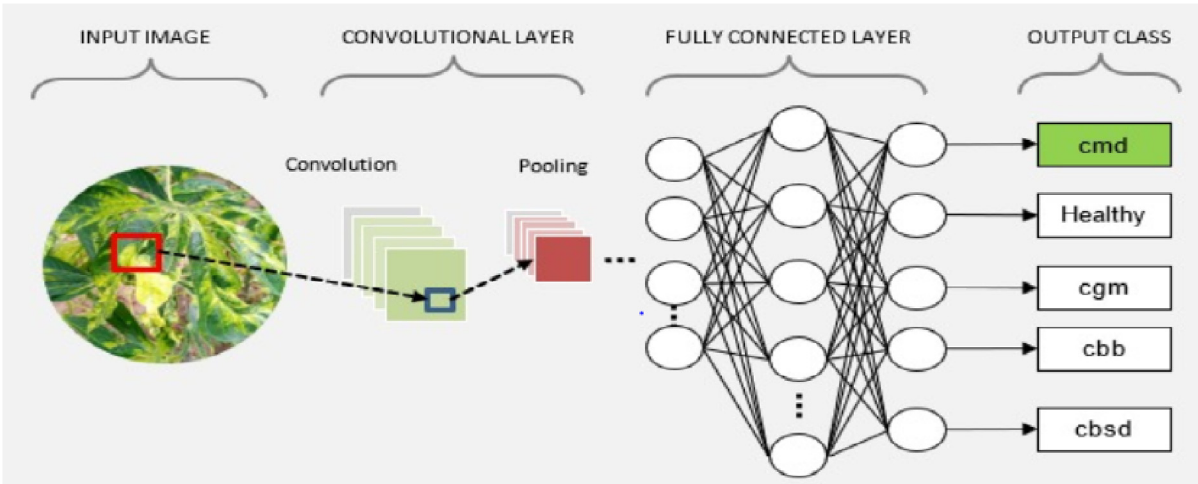


Fig. 2. The model architecture.

ference as illustrated in Fig. 5. In Table 1, Image number for each category of cassava leaf in training set were shown.

These images were highly imbalanced being heavily biased towards cmd and cbsd classes as shown in Figs. 3 and 4 below.

To address the above issues, the following steps were taken:

i) We tried to improve image contrast by using CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm [24,25] CLAHE can help computer vision algorithms to perform much better in low resolution and poor contrast [26,27,6].

ii) Label skew: put it that you have a dataset at your reach, whereby you can train a machine learning algorithm to reach an accuracy of 97%. Overly excited, you then convince technocrats to deploy the model because who would even refuse such a model with that kind of amazing performance, unfortunately, the model fails to perform in real world. Why? Because there are many conditions that can lead to this and one that is very common is class imbalance and there must be techniques to counter such. Balancing is not easy, for our case, we used a combination of techniques that are already in literatures: class-weight, focal loss and SMOTE coupled with data augmentation techniques to increase the size of training set which gave an improved accuracy. SMOTE [14,15] aims to balance class distribution through randomly increasing the number of minority class representation by replicating them. Generation of virtual training data points occurs by mean of interpolation of minority class such that minority class set B, for each $R \in B$, the k-nearest neighbors of R are obtained by calculating the Euclidean distance between R and every other sample in set B. The sampling rate Z is set according to imbalance proportion, such that for each $R \in B, Z$ data points $(thatis, r_1, r_2, \cdots r_n)$ are randomly selected from its k-nearest neighbors, and they construct a set $B1$. Finally, for each data point $R_k \in B_1 (k = 1, 2, 3 \cdots N)$, the formula $R' = R + r \ and \ (0.1) * |R - R_k|$ generates a new data point where rand (0, 1) represents the random number between 0 and 1.

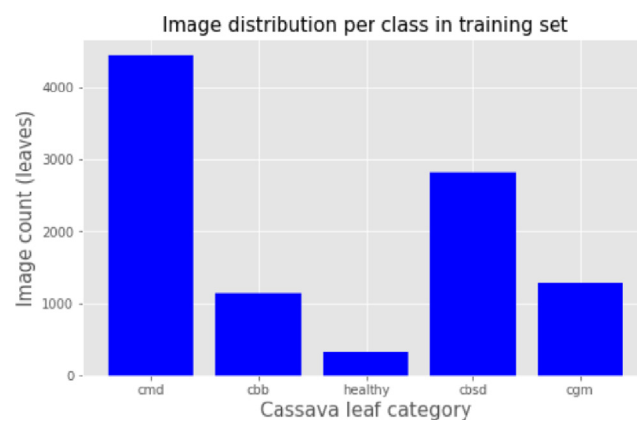


Fig. 3. Bar chart showing the number of samples per class (unbalanced dataset).

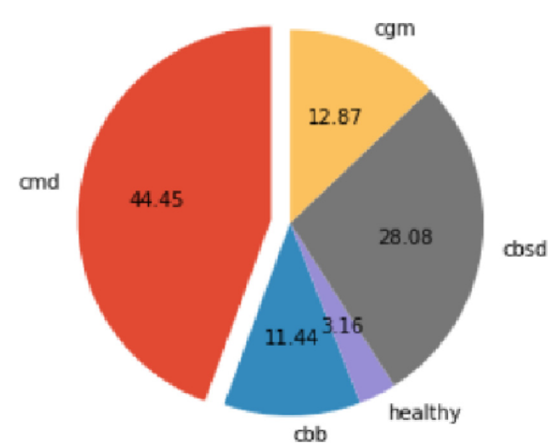


Fig. 4. Pie chart showing the number of samples per class (unbalanced dataset). The top class has 44.45% while the least represented class has 3.16%, revealing an order of magnitude difference.

Table 1
Image number for each category of cassava leaf in training set.

Category	cmd	cbsd	cgm	healthy	cbb
Image number	4445	2808	1287	3 16	1144

iii) Artificially increasing the size of the image dataset through image flipping, random shearing, random cropping, random scaling, center zooming, height and width shift were the augmentation techniques employed to remediate the problem of the dataset is small in size. By increasing the size of the dataset through the image flipping technique, it will be more helpful in training and testing, so that it will give more accuracy.

During training, there were two types of parameters: the parameters that were learned from the model and these were the weights and some other parameters that were being tuned and these were hyper-parameters, for example, learning rates, the number of epoch, batch size, input shape, optimizer and the number of neurons in the hidden layers. In Table 2, Effect of different input image dimensions on CNN model accuracy were discussed. So, when CNN [20] was training, it was trained with some of these hyper-parameters, however, the model was improved by finding some of the best values of these hyper-parameters through tuning techniques. One of the main hyperparameters tuned was the learning rate where we used Cyclic Learning Rate (CLR) [32], the cycle consists of two kinds of steps; one step that increases linearly from minimum to maximum and the other that decreases linearly. The tuning was done through Learning Rate Finder (LRF) that basically tested several combinations of these values and eventually returned the best learning rate [minimum learning rate, maximum learning rate] for the model as shown in Fig. 6. We used a Cyclic Learning Rate because of the concept of super-convergence [33], the use of large learning rates regularizes the network which results in the reduction of all other kinds of regularization to keep a balance between overfitting and underfitting. The goal was also to maximum performance by minimizing the computational time required since the larger dimension image was being used. Other hyper-parameters selections and best choices were achieved through grid search with k-fold cross validation where k = 3. The reason being neural networks are difficult to configure and there are a lot of parameters that need to be set up beside, individual models can be very slow to train.

5. Results and discussions

5.1. LRF

Observing Fig. 5 below, we can see that our network was able to start learning at around 1e-7, from 1e-10 to around 1e-7, the learning rate was too low and network unable to learn. The lowest lost was found at 1e-3 after which loss started to increase sharply up to 1e-2 then decreased to 1e-1 from which it finally exploded, meaning the learning rate was too large for the network to learn anymore. Precision, Recall, F1-Measure and accuracy are shown in Eqs. (1)–(4). Precision is the measure of accurately predicted true positive values to the total number of positive predicted observations (Eq. (1)) [37]. Recall is the measure of number of positive class predictions made with the all positive predictions (Eq. (2)). F-Measure is the measure which balances both the precision and recall (Eq. (3)).

Table 2
Effect of different input image dimensions on CNN model accuracy.

Input Shape	Accuracy (%)	Loss (%)	Epochs	Time per Epoch
(128,128,3)	76.9	26	124	699 s
(224, 224,3)	80	18	124	1513 s
(256, 256, 3)	89.0	10	124	2065 s
(448, 448, 3)	93.0	0.06	124	3600 s

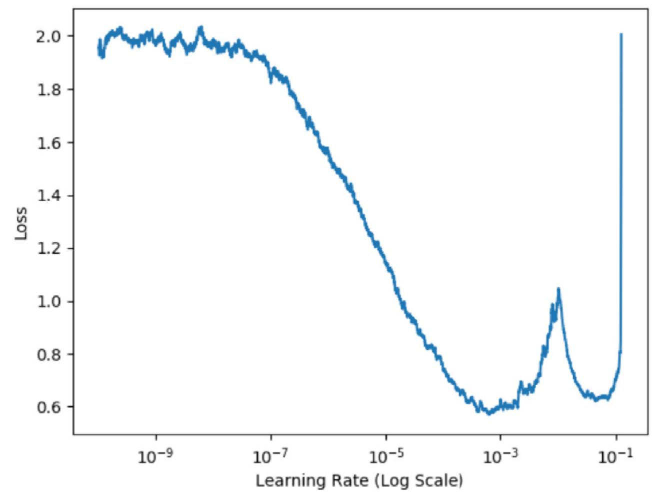


Fig. 5. Learning rate finder Cyclic Learning Rate; minimum learning rate boundary was 1e-7 and the maximum learning rate boundary was 1e-3 for our dataset.

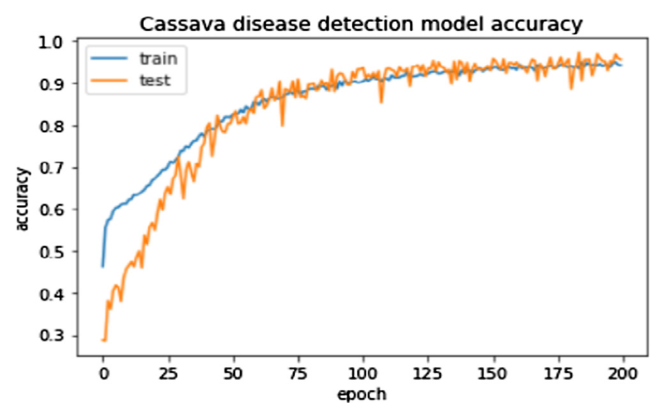


Fig. 6. Cassava disease detection model accuracy.

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \quad (1)$$

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \quad (2)$$

$$\text{F - Measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

$$Accuracy = \times \frac{True\ \ Positive + True\ \ Negative}{True\ \ Positive + True\ \ Negative + False\ \ Positve + False\ \ Negative}$$

Performance Evaluation: In Data Science, evaluating model performance is so important that the most commonly used performance metrics in classification are; confusion matrix [normalized, non-normalized], accuracy, precision [38,39], sensitivity (recall),

Table 3
Classification report before the focal loss, class weight, and SMOTE were implemented.

Image classes	Precision	Recall	F1-Score	Support
cbb	0.81	0.83	0.82	466
cbsd	0.92	0.91	0.92	887
cgm	0.80	0.72	0.76	447
cmd	0.97	0.96	0.97	1719
healthy	0.70	0.67	0.69	241

specificity, F1 score, Precision-Recall curve and AUC-ROC (Area Under The Curve – Receiver Operating Characteristics curve) [37,40]. AUC-ROC is mainly used to evaluate the model performance of a balanced dataset while the Precision-Recall curve is used for imbalanced dataset evaluation. Accuracy is mostly used to judge performance of a model; however, it suffers anomaly when classes are imbalanced, when we take for example a cancer detection, the chances of having a cancer is really low in that out of 1008 patients only 8 have cancer meaning there is a high chance that the model can detect everyone as not having cancer with 99% accuracy missing out on patients having cancer that goes undetected – recall. Meaning we need other alternative methods as well to supplement the accuracy metric. In this study, we evaluated the model using a classification report besides the accuracy metric to evaluate performance on a per-class analysis basis as illustrated by Tables 3 and 4. In Fig. 6:Cassava disease detection model accuracy were shown. In Fig. 7: Accuracy and Mean Squared Error for CNN after applying techniques to counter the class imbalance were shown.

Table 4
Classification report when focal loss, class weight, and SMOTE were implemented.

Image classes	Precision	Recall	F1-Score	Support
cbb	0.93	0.91	0.92	466
cbsd	0.93	0.92	0.93	887
cgm	0.90	0.89	0.90	447
cmd	0.96	0.94	0.95	1719
healthy	0.92	0.91	0.92	241

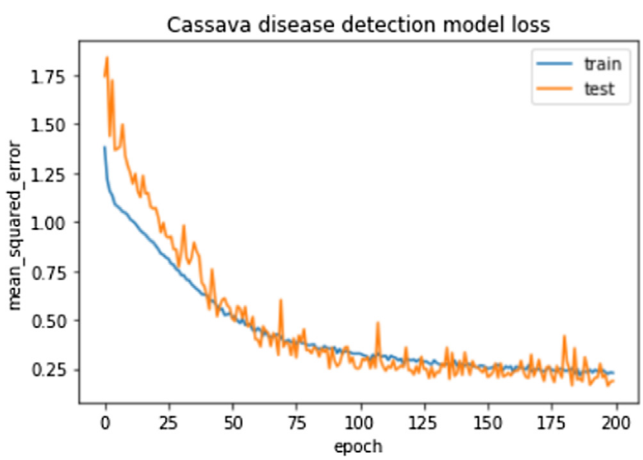


Fig. 7. Accuracy and Mean Squared Error for CNN after applying techniques to counter the class imbalance. The accuracy achieved was above 93% with loss below 10% though learning was a bit volatile.

From Fig. 8 Cyclic Learning Rate of the model for 164 epochs using a "triangular" policy were discussed. First, the model didn't overfit as the train and validation accuracy are close and following each other. Secondly, the model could probably be trained a little more as the trend for accuracy on both training and validation datasets still rising for the last few epochs. Therefore, the model has not yet overlearned the training dataset, showing reasonable skill on both datasets.

When we closely look at Table 4, we can see that the model is good at predicting CMD, CBSD, CBB and healthy classes with a precision of 96%, 93%, 93%, and 92% respectively. However, the model performed poorly on predicting CGM class with 90% precision, visually CGM looks closer to CMD. probably some CGM were misclassified under CMD.

5.2. Precision, recall, F1-Score

Precision-Recall is a very useful measure of success of prediction when classes are very imbalanced and in Table 4 is a classification report from the experiment conducted when techniques to take care of class imbalance were implemented. Table 3 is a classification report from the experiment without techniques to counter the class imbalance.

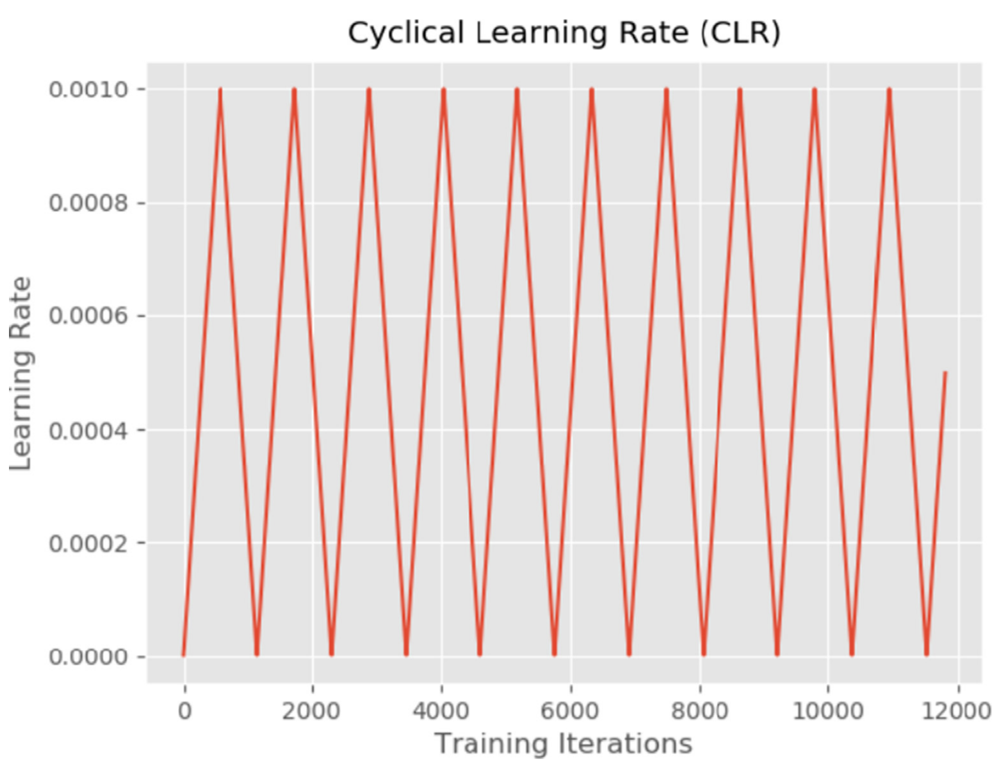


Fig. 8. Cyclic Learning Rate of the model for 164 epochs using a "triangular" policy.

Shown below in Fig. 9 are randomly predicted images from test data (these images were used neither in training nor validation process). The model is good at predicting cassava disease categories.

Fig. 9. Randomly predicted images by the model from the test set (these were images neither used in training nor validation data).

6. Conclusions

In this work, the CNNs model was developed and trained with a very limited dataset having high class imbalanced. The data was heavily biased towards CMD and CBSD classes, hence, necessitated for various techniques to counter class imbalanced. Our model performance showed promising results and to the best of our knowledge, there is still no experimentation done for classification of cassava mosaic and other cassava disease detection by training CNNs from scratch using an imbalanced dataset. Techniques applied were class weight, focal loss, SMOTE and different image dimensions which we found input shape of a vector (448, 448, 3) giving the best performance. Therefore, this model can be integrated into mobile applications for use by farmers or agricultural extension workers in mobile devices such as smartphones or other unmanned aerial vehicles to be used for real-time monitoring and early warning of cassava infections so that necessary prevention methods be applied or large scale implementation can be deployed via satellite imagery analysis to detect areas with infestation and recommend prevention techniques since most subsistence farmers in Sub-Saharan Africa cannot afford smartphones due to low level of income.

We found it very interesting how the model performance for the imbalanced dataset can be highly improved by class-weight, SMOTE, focal loss techniques and large input shape dimensions of images. From this experiment, we got an increment of over 5% in accuracy and a log loss that dramatically reduced to 0.06% from over 20% when class-imbalanced rectification techniques coupled with data augmentation and large input image dimensions were used. One key area that still needs further research though is multiple co-occurring diseases on the same plant.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Kehinde AT. Utilization potentials of cassava in Nigeria: the domestic and industrial products. Food Rev Intl 2006;22:29–42.
[2] Aduni UA, Ajayi OA, Bokanga M, Dixon BM. The use of cassava leaves as food in Africa. Ecol Food Nutr 2005;44:423–35.
[3] Sladojevic S, Arsenovic M, Anderla A, Culibrk D, Stefanovic D. Deep neural networks based recognition of plant diseases by leaf image classification. Comput Intellig Neurosci 2016;2016:11.
[4] Huang KY. Application of artificial neural network for detecting Phalaenopsis seedling diseases using color and texture features. Comput Electron Agric 2007;57:3–11. doi: https://doi.org/10.1016/j.compag.2007.01.015.
[5] Price TV, Gross R, Wey JH, Osborne CF. A comparison of visual and digital image-processing methods in quantifying the severity of coffee leaf rust (Hemileia vastatrix). Aust J Exp Agric 1993;33:97–101. doi: https://doi.org/10.1071/EA9930097.
[6] Navneet Dalal, Bill Triggs. Histograms of Oriented Gradients for Human Detection. In: CVPR, pages 886–893, 2005.
[7] Singh Vijai, Misra AK. Detection of plant leaf diseases using image segmentation and soft computing techniques. Inform Process Agric 2016;4. doi: https://doi.org/10.1016/j.inpa.2016.10.005.
[8] Fuentes A, Yoon S, Kim S, Park D. A robust deep-learning-based detector for real-time tomato plant diseases and pest's recognition. Sensors 2017;17(9):2022.
[9] Ramcharan A, Baranowski K, McCloskey P, Ahmed B, Legg J, Hughes DP. Deep learning for image-based cassava disease detection. Front Plant Sci 2017;8:1852.
[10] Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 2018;145:311–8.
[11] LeCun Y, Bottou L, Orr G, Mller K. Efficient backprop. Neural Networks: Tricks of the trade. Springer; 1998.
[12] Wang S, Yao X. Using class imbalance learning for software defect prediction [J]. IEEE Trans Reliab 2013;62(2):434–43.
[13] Lin T-Y, Goyal P, Girshick RB, He K, Dollár P. Focal loss for dense object detection. In: In: IEEE international conference on computer vision (ICCV). p. 2999–3007.

[14] Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: Synthetic minority over-sampling technique. J Artif Intell Res 2002;16:341–78.

[15] Krawczyk B. Learning from imbalanced data: open challenges and future directions. Prog Artif Intell 2016;5(4):221–32. doi: https://doi.org/10.1007/s13748-016-0094-0.CrossRefGoogle Scholar.

[16] F. Chollet, Keras, https://keras.io/, 2015.

[17] Kaggle competition, https://www.kaggle.com/c/cassava-disease/overview, 2019.

[18] Legg JP, Thresh JM. Cassava mosaic virus disease in East Africa: a dynamic disease in a changing environment. Virus Res 2000;71:135–49.

[19] Godliver O, Friedrich M, Mwebaze E, Quinn John A, Biehl M. Machine Learning for diagnosis of disease in plants using spectral data. In: Int'l Conf. Artificial Intelligence, ICAI'18.

[20] Krizhevsky A, Sutskever I, Hinton GE, Imagenet classification with deep convolutional neural networks, in Advances in Neural Information Processing Systems; 2012.

[21] Ciresan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. In: CVPR, 2012.

[22] Demir-Kavuk O, Kamada M, Akutsu T, et al. Prediction using step-wise L1, L2 regularization and feature selection for small data sets with a large number of features. BMC Bioinf 2011;12:412. doi: https://doi.org/10.1186/1471-2105-12-412.

[23] Xu ZongBen ZH. Wang Yao, Chang XiangYu, Yong Liang: L1/2 regularizer. Sci China 2010;53(6):1159–69.

[24] Hummel RA. Image enhancement by histogram transformation. Comp Graphics Image Process 1977;6:184195.

[25] Zuiderveld K. Contrast limited adaptive histogram equalization. In: Heckbert P, editor. Graphics gems IV. Academic Press; 1994. ISBN 0-12-336155-9.

[26] Ketcham DJ, Lowe RW, Weber JW. Image enhancement techniques for cockpit displays. Tech. rep., Hughes Aircraft; 1974.

[27] He L, Luo L, Shang J. An image enhancement algorithm based retinex theory. first international workshop on education technology and computer science; 2009, 3, 350–352.

[28] Onzo Alexis, Hanna Rachid, Sabelis M. Biological control of cassava green mites in Africa: impact of the predatory mite/typhlodromalus aripo. J Organic Chem 2005:2–7.

[29] Legg JP, Jeremiah SC, Obiero HM, Maruthi MN, Ndyetabula I, Okao-Okuja G, et al. Comparing the regional epidemiology of the cassava mosaic and cassava brown streak virus pandemics in Africa. Virus Res 2011;159(2):161–70.

[30] Hillocks Rory, Thresh JM. Cassava mosaic and cassava brown streak virus diseases in Africa: a comparative guide to symptoms and aetiologies. Roots 1998;7.

[31] Otim-Nape GW, Alicai T, Thresh JM. Changes in the incidence and severity of cassava mosaic virus disease, varietal diversity and cassava production in Uganda. Ann Appl Biol 2001;138(3):313–27.

[32] Leslie N. Smith. Cyclical learning rates for training neural networks. In Proceedings of the IEEE Winter Conference on Applied Computer Vision; 2017.

[33] Leslie N. Smith. Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates. In: arXiv:1708.07120v3 [cs. LG] 17 May 2018.

[34] Bock CH, Poole GH, Parker PE, Gottwald TR. Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. In: Critical reviews in plant sciences, Volume 29, 2010 – Issue 2, Pages 59–107.

[35] Hillocks R, Raya M, Thresh J. The association between root necrosis and above-ground symptoms of brown streak virus infection of cassava in southern Tanzania. Int J Pest Manage 1996:285–9.

[36] Abdullahi I, Atiri G, Dixon A. Effects of cassava genotype, climate, and the Bemisia tabaci vector population on the development of African cassava mosaic geminivirus (acmv). Acta Agronom Hungarica 2003:285–9.

[37] Ullah Irfan et al. A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. IEEE Access 2019:60134–49.

[38] Sharma Manik, Sharma Samriti, Singh Gurvinder. Performance analysis of statistical and supervised learning techniques in stock data mining. Data 2018;3(4):54.

[39] Hossin Mohammad, Sulaiman MN. A review on evaluation metrics for data classification evaluations. Int J Data Min Knowledge Manage Process 2015;5(2):1.

[40] Kaur Prableen, Sharma Manik. Diagnosis of human psychological disorders using supervised learning and nature-inspired computing techniques: a meta-analysis. J Med Syst 2019;43(7):204.