

# CassavAI: Cassava Leaf Disease Detection using Neural Networks

Dennis Dominic S. Diego<sup>1</sup>, Jason A. Dolorso<sup>1</sup>, Cymon P. Marcaida<sup>1</sup>, Naman Punit<sup>1</sup>, Matthew C. Romero<sup>1</sup>

<sup>1</sup>Aboitiz School of  
Innovation, Technology,  
and Entrepreneurship,  
Asian Institute of  
Management, Paseo de  
Roxas, Makati City,  
Philippines

January 2021

## Abstract

In the Philippines, Cassava is considered one of the staple crops because of its ease in cultivation and profitability. However, the threat of various diseases hampers the growth and volume of cassava harvest which could be used for livelihood or exportation. It is imperative then that the diseases be detected at an early stage. This however falls into a subjective factor, as farmers or scientists often rely on optical inspection and domain knowledge whether a specific disease is one or the other. Since most cassava diseases manifest on leaves, we propose a deep learning method to detect cassava leaf diseases using neural networks, specifically pre-trained neural networks. Utilizing transfer learning on an EfficientNetB3 variant and training on 10,791,220 parameters, our model was able to achieve an accuracy of 88.60%. After applying data augmentations and ensemble methods, we were able to achieve an accuracy of 89.90%, beating the baseline proportional chance criterion of 50.46%. Despite the imbalanced data, we were able to create a neural network classifier that could accurately classify cassava leaf diseases which could potentially help farmers and agricultural scientists alike. In this project, we also outlined our recommendations to further the efforts in classifying plant leaf diseases and advance smart agriculture in the country.

**Keywords:** Cassava disease detection; Convolutional neural networks; Efficientnet; Transfer learning

## 1 Introduction

Cassava (*Manihot esculenta*) is considered one of the top crops in the Philippines. Some parts of the plant can be utilized for various purposes, like the leaves, which can be eaten raw, or the roots because of its starchy properties be also eaten as it is a good source of energy [1]. Due to the ease of farming and profitability, it is no surprise that it is a staple in the Philippine agricultural landscape, so much so that a National Cassava Roadmap was put in place to bring cassava harvesting to an optimal level [2].

A major challenge in the cultivation of cassava root crops is the existence of various diseases. As much as Cassava is easily cultivated, it is very susceptible to multiple diseases [3]. It is imperative then that early detection of these diseases is prioritized.

Plant and crop disease diagnosis is often done through optical observation and noting several symptoms on plant leaves. This, however, entails a high degree of complexity. The difference between leaf appearance and disease manifestations especially in Cassava, where there are at least a number of different diseases, existing phytopathological problems may be hard to detect even for a trained eye [4]. As it is, plant crop disease diagnosis remains subjective, and the technology to maybe hasten the diagnosis is often limited [5].

Better approaches are needed to aid especially farmers to save their crops when a disease affects cassava crops. One way to do this is by leveraging on automation and the use of data science. With this in mind, we propose a deep learning approach to detecting diseases in cassava leaves with the utilization of pre-trained convolutional neural networks. This work presents our techniques in using CNNs on an imbalanced dataset, which was still able to achieve an accuracy of at least 89%, beating a baseline accuracy score of 50% proportional chance criterion. Our methodology is outlined in the next sections.

## 2 Related Works

Applications of Deep Learning and Convolutional Neural Networks have been gaining traction in the past decade. As early 2010, [6] highlighted the need to develop a rapid, cost-effective, and reliable health-monitoring system to promote advancements in agriculture. This trend was further accelerated by the launch of ImageNet Large Scale Visual Recognition Challenge (ILSVRC) during the same year. It is an annual challenge to evaluate algorithms for large-scale object detection and image classification that paved the way for state-of-the-art algorithms to be developed year after year [7]. Some of the notable architectures are Inception (or GoogleNet), which showed that CNN layers do not always have to be sequential [8], Residual Network (ResNet) that introduced residual block to reduce overfitting [9]; and VGG that used a large number of stacks and smaller size filters [10].

These architectures' influence widened as they can be implemented, through transfer learning, in different applications — one of which is in plant disease recognition. A systematic review in 2020 revealed that 85% of the 121 literatures about plant disease classification used transfer learning, fine-tuning, and hyperparameter methods to improve their models [11].

Excluding some large architectures requiring industrial-grade computing power, research involving plant disease detection using CNN often utilized architectures/models of past ILSVRC winners. Said models are an obvious choice as they are considered the best implementation during their time. Within the context of Cassava leaves disease detection, Ramcharan et al. used Inception as their base model that achieved 93% accuracy in classifying five diseases [12].

As of writing, 7 of the top 10 models in terms of ImageNet top-1 accuracy are based on the EfficientNet developed by Tan et al. from Google [13]. EfficientNet is a CNN architecture and scaling method that uses a *compound coefficient* to uniformly scale all dimensions of depth, width, and resolution, leading to better accuracy and efficiency than other architectures like ResNet and Inception-V2 [14].

### 3 Methodology

The algorithm used for this study is Convolutional Neural Networks with the implementation steps in the sequence of data collection, image augmentation, transfer learning, model tuning, cross-validation ensembling, and test time augmentations. The model will then be evaluated through the Kaggle public leaderboards containing 15,000 test images and an out-of-fold dataset. Model training will use the TensorFlow 2.0 API in a 2080ti GPU with a memory of 11 GB.

#### 3.1 Data Collection

The data used in this study came from a Kaggle competition *Cassava Leaf Disease Classification*, sponsored by Makerere University AI Lab — a research group based in Uganda. The dataset contains 21,367 labeled images of Cassava leaves that were annotated with National Crops Resources Research Institute's help, also located in Uganda [15].

The data contains five classes. These are healthy cassava leaves and leaves with four types of diseases: Bacterial Blight, Brown Streak Disease, Green Mottle, and Mosaic Disease. The dataset is imbalanced, with 61.50% of the images labeled with Mosaic Disease. Images are captured in their natural environment (as opposed to laboratory-grade photos). Some photos include noise like the ground, other plants, and even human legs, as seen in Figure 1.



Figure 1 Sample Cassava Leaves Diseases. From left to right, top to bottom: (1) Healthy, (2) Mosaic Disease, (3) Brown Streak Disease, (4) Bacterial Blight, (5) Green Mottle.

#### 3.2 Data Augmentation

Data Augmentation is a widely used technique, especially for image classification problems, as it helps improve the performance and ability of a model to generalize through expanding the training dataset. One instance would be AlexNet that utilized random cropping and horizontal flipping for evaluation on the CIFAR dataset.

To augment the images of Cassava leaves, we used random cropping of 512x512, random horizontal and vertical flipping, random rotation with a range of  $\pm 0.2 * 2\pi$ , random zoom of up to 20%, and random contrast change from -0.2 to 0.2, all of which are through TensorFlow's preprocessing API.

### 3.3 Transfer Learning

Building and training a CNN architecture from scratch would be time-consuming, resource-extensive, and require numerous experimentations and tuning. Instead, we employed transfer learning, which takes a model trained on a large dataset and transfer its learning knowledge into our dataset.

As discussed in section 2, most of the state-of-the-art models are based on the EfficientNet architecture. To balance depth and computational efficiency, we used the EfficientNetB3 variant with ImageNet weights. Global average pooling, dropout, and dense with softmax function is then added on top. Table 1 shows a summary of the layers and their respective parameters.

Layer (type)	Output Shape	Param #
Input Layer	None, 512, 512, 3	0
Augmentations	None, 512, 512, 3	0
EfficientNetB3	None, 16, 16, 1536	10,783,535
Global Average Pooling 2D	None, 1536	0
Dropout	1536	0
Dense	None, 5	7685

Table 1 Model Architecture

Overall, the model has a total parameter of 10,791,220, of which 10,703,917 are trainable, and the remaining 87,303 are non-trainable.

### 3.4 Model Tuning

While exploring the data, we found out that the training set is noisy; some images are mislabeled. Figure 2 shows some examples of leaves labeled as “healthy” that we suspect are incorrectly tagged due to apparent signs of disease.



Figure 2 Images incorrectly tagged as healthy.

We applied label smoothing to be able to work on a dataset with noisy labels and enhance the model’s generalization. Label smoothing is the modification of the loss function to prevent the network from becoming *over-confident* with the current dataset [16]. Specifically, we extended the Sparse Categorical Cross-Entropy (SCCE) loss function to add a label smoothing of 0.3.

Adam optimizer was used as it fits well with noisy gradients and models with many parameters [17]. We also implemented cosine decay to schedule the change of learning rate wherein the initial learning rate was set to 0.0001 and would eventually decrease.

### 3.5 Cross-Validation Ensemble

Cross-validation is a resampling procedure that is primarily used for the evaluation of machine learning models. But to make our prediction for cassava disease more robust, we used k-fold cross-validation for ensembling five models of the same architecture but trained on different train/test splits.

This methodology is similar to *Bagging*, introduced [18]. It is the process of generating multiple versions of a predictor and using them to get an aggregated predictor.

We first set aside an out-of-fold set comprised of 20% of the images to be used for evaluation. Five cross-validation folds are then used for the other 80% of the dataset using the same model architecture developed. All five models produced will then be used for inferencing.

### 3.6 Test Time Augmentation

Data augmentation, same as those applied in section 3.2, can also be used for the inference phase to yield better predictive performance. In [10], the authors also applied horizontal flip test-time augmentation that produced state-of-the-art results during that time on the ILSVRC dataset.

The test time augmentations we experimented with are center cropping, horizontal and vertical flipping, random zooming, random contrast, and normalization. Every image will go through four times test time augmentations.

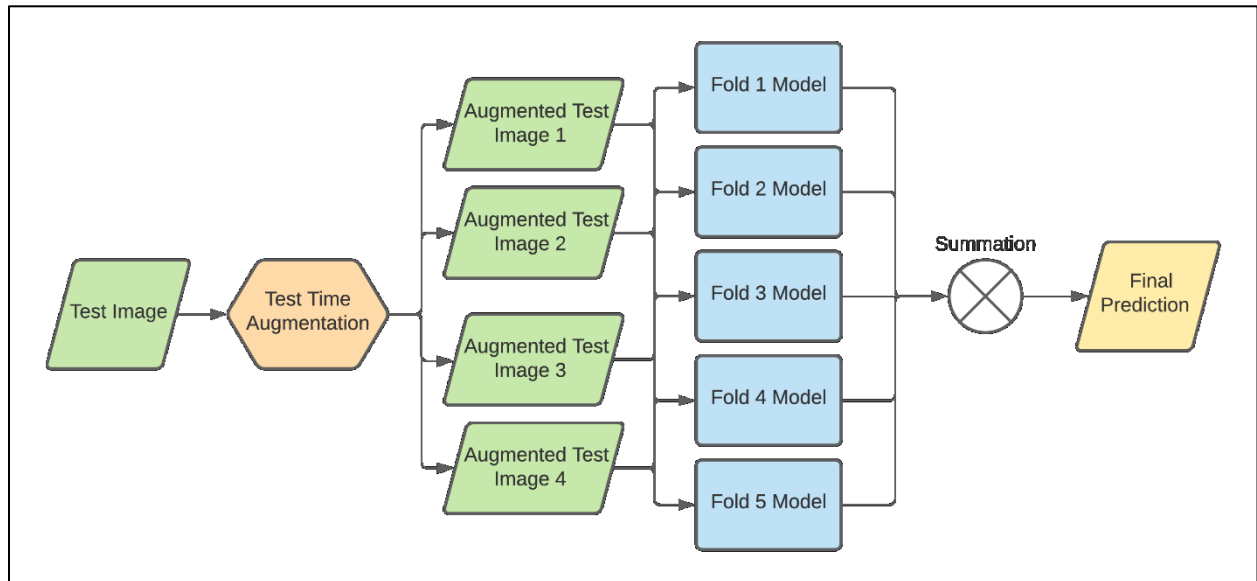


Figure 3 Inference Process Flow Diagram

Figure 3 shows the entire inference process for a single image. Augmentations will be applied to produce four randomly augmented images. All five models from the cross-validation are used to create an ensemble where each of their predictions will be summed up, and the class with the greatest probability would come up as the final prediction.

## 4 Results and Evaluation

In evaluating the model's performance, we used the out-of-fold data and the Kaggle competition's public leaderboard.

### 4.1 Kaggle Competition Leaderboards

As of this paper's writing, the competition that served as the data source is still on-going in Kaggle, joined by 3,016 teams with up to 5 members comprised of data scientists worldwide. On top of the provided images for training, the organizer set aside 15,000 images that are hidden to the participants to determine who has the best model. The participants can then submit their models to be evaluated on 31% of hidden images. The best score of each team will be posted on a public leaderboard. When the contest ends, rankings will be determined on 100% of the hidden images.

To give us an idea of how our model fares, we submitted our model to the competition and checked the accuracy through the public leaderboard.

#### 4.1.1 Label Smoothing

We first tried out single models without cross-validation ensembling to find out the best value for label smoothing.

Model	Label Smoothing	Test Time Augmentations	TTA Runs	Accuracy
EfficientNetB3	0	Flip, Zoom, Contrast	4	87.60%
EfficientNetB3	0.1	Flip, Zoom, Contrast	4	88.40%
EfficientNetB3	0.3	Flip, Zoom, Contrast	4	<b>88.60%</b>

Table 2 Single Model Experiments with Label Smoothing

Table 2 shows the results of different label smoothing values, with 0.3 as the ceiling, since it will trim down the confidence of the model's prediction by as high as 30%. Based on the results, label smoothing of 0.3 produced the best accuracy —evidence of how noisy the labels are.

#### 4.1.1 Ensemble and Test Time Augmentations

We explored different test time augmentations (TTA) on 5 fold ensemble and 4 TTA runs for our next experiments.

Model	Folds	Test Time Augmentations					TTA Runs	Accuracy
		Flip	Zoom	Contrast	Center Crop	Normalize		
EfficientNetB3	5	h/v	-0.2, 0.2	0.2, 0.2	n/a	False	4	89.40%
EfficientNetB3	5	h/v	-0.3, 0.3	0.5, 0.5	n/a	False	4	89.50%
EfficientNetB3	5	h/v	-0.3, 0.3	0.5, 0.5	n/a	True	4	89.20%
EfficientNetB3	5	h/v	-0.3, 0.3	0.5, 0.5	512x512	False	4	89.40%
EfficientNetB3	5	h/v	-0.2, 0.2	0.2, 0.2	512x512	True	4	89.30%
EfficientNetB3	5	h/v	-0.3, 0.3	0.5, 0.5	512x512	True	4	<b>89.60%</b>

Table 3 5-fold experiments with different Test Time Augmentations

Table 3 shows that all models involving cross-validation ensembling perform better than just a single model. The best result is achieved by using higher contrast and zoom range while also adding center cropping and normalization.



## 4.2 Out-of-fold Metrics

Despite a large number of test images available through the Kaggle competition leaderboard, it only provides accuracy. It lacks other metrics like precision, recall, and F1-score, to prevent participants from tampering with the test set and “gaming” the system.

The out-of-fold (OOF) data as described in section 3.5 will be used to obtain other evaluation metrics. The results for the OOF is expected to slightly vary with the values from the Kaggle leaderboards considering that the metrics are based on a different test set and the training images used for OOF is 20% less. The baseline accuracy is the proportional chance criterion (PCC)\*1.25 of 50.46%.

	precision	recall	f1-score	accuracy
<b>Bacterial Blight (CBB)</b>	75.76%	59.52%	66.67%	<b>89.90%</b>
<b>Brown Streak Disease (CBSD)</b>	84.72%	84.14%	84.43%	
<b>Green Mottle (CGM)</b>	83.41%	78.02%	80.62%	
<b>Mosaic Disease (CMD)</b>	95.92%	97.61%	96.76%	
<b>Healthy</b>	72.92%	77.46%	75.12%	
<b>macro avg</b>	82.55%	79.35%	80.72%	
<b>weighted avg</b>	89.76%	89.90%	89.76%	

Table 4 Out-of-Fold Classification Report

Table 4 shows that the accuracy of the 5-fold ensemble with the out-of-fold test set is consistent with that in the Kaggle Leaderboards, with just a difference of 0.3%.

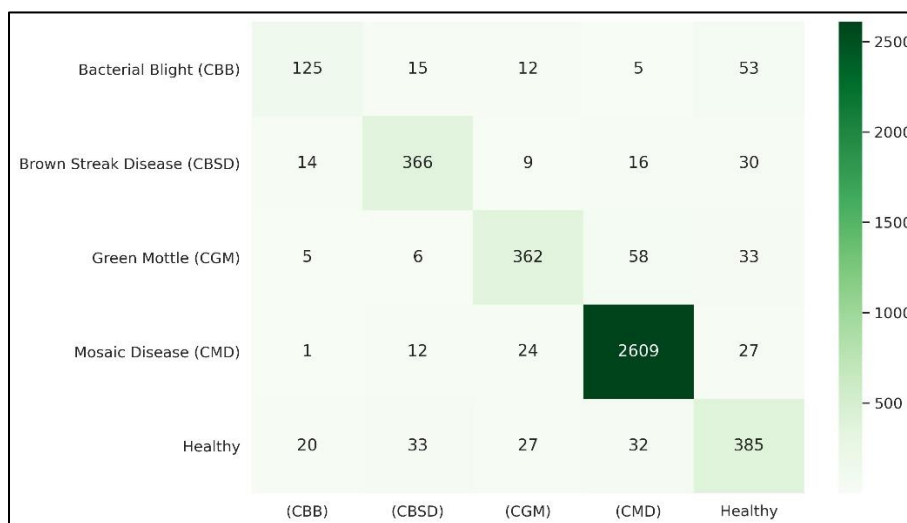


Figure 4 Out-of-Fold Confusion Matrix

As mentioned in section 3.1, the dataset is imbalanced, with 61.50% labeled as CMD — this is consistent with the confusion matrix being heavy on the same class. Still, the model was able to perform well, beating the PCC baseline of 50.46%. Considering that the images used in our training were raw and of low quality, the accuracy is not far off the 93% accuracy of Ramcharan et al. in [12], which used photos captured using a 20.2-megapixel digital camera and manually cropped to a single leaflet.

## 5 Conclusion & Recommendation

Our model's results proved that transfer learning with convolutional neural networks is a robust method to identify different cassava leaf diseases, and can also be applied extensively in the agriculture industry. For example, the model could be integrated in smart farming where drones with video capabilities could be used to detect whether cassava leaves has a disease or not in real-time. This will also democratize the use of AI for smart farming to both current and future cassava farmers and help them in preventing diseases and optimizing their yields. Furthermore, the model could also augment the cassava roadmap of the Department of Agriculture especially for its programs related to diseases detection and prevention.

It should be noteworthy that only four cassava leaf diseases were included in the dataset, which means there's a lot of possibilities that our model can be further improved. This includes classifying beyond the four leaf diseases and incorporate other conditions such as the cassava phytoplasma disease (CPD). Besides detecting leaf diseases in Cassava, our model could be further improved by being able to detect leaf diseases from other crops such as corn or bananas, or extending its capability to other parts of the crop as well.

### Conflict of Interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the factors, knowledge or materials discussed in this project.

### Acknowledgments

We would like to thank our mentors, Prof. Chris Monterola, Prof. Erika Legara, Prof. Carlo Vincienzo Dajac, and the rest of ACCeSs lab data scientists for their guidance.

### Author's contributions

All members of the team designed and performed the study, analyzed the results, and authored the manuscript.

### References

- [1] Taiwo, Kehinde A. (2006). Utilization Potentials of Cassava in Nigeria: The Domestic and Industrial Products. *Food Reviews International*, 22(1), 29–42.  
doi:10.1080/87559120500379787
- [2] Taiwo, Kehinde A. (2006). Utilization Potentials of Cassava in Nigeria: The Domestic and Industrial Products. *Food Reviews International*, 22(1), 29–42.  
doi:10.1080/87559120500379787
- [3] Okogbenin, E., Egesi, C. N., Olasanmi, B., Ogundapo, O., Kahya, S., Hurtado, P., ... & Fregene, M. (2012). Molecular marker analysis and validation of resistance to cassava mosaic disease in elite cassava genotypes in Nigeria. *Crop Science*, 52(6), 2576-2586.



- [4] Ferentinos, Konstantinos P. (2018). Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, 145, 311–318. doi:10.1016/j.compag.2018.01.009
- [5] Bock, C., Poole, G., Parker, P., and Gottwald, T. (2010). Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. *Crit. Rev. Plant Sci.* 29, 59–107. doi: 10.1080/07352681003617285
- [6] Sankaran, S., Mishra, A., Ehsani, R., & Davis, C.E. (2010). Review: A review of advanced techniques for detecting plant diseases. *Computers and Electronics in Agriculture*, 72, 1-13.
- [7] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 211-252.
- [8] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1-9.
- [9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778.
- [10] Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.
- [11] Abade, A., Ferreira, P., & Vidal, F. (2020). Plant Diseases recognition on images using Convolutional Neural Networks: A Systematic Review. *ArXiv*, abs/2009.04365.
- [12] Ramcharan, A., Baranowski, K., McClowsky, P., Ahmed, B., Legg, J., & Hughes, D. (2017). Deep Learning for Image-Based Cassava Disease Detection. *Frontiers in Plant Science*, 8.
- [13] Papers with Code. (n.d.) *Image Classification on ImageNet*. Retrieved January 25, 2021, from [www.paperswithcode.com/sota/image-classification-on-imagenet](http://www.paperswithcode.com/sota/image-classification-on-imagenet).
- [14] Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv*, abs/1905.11946.
- [15] Makerere University AI Lab. (2020). Cassava Leaf Disease Classification. Retrieved December 20, 2020, from <https://www.kaggle.com/c/cassava-leaf-disease-classification/data>.
- [16] Müller, R., Kornblith, S., & Hinton, G.E. (2019). When Does Label Smoothing Help? *NeurIPS*.
- [17] Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. *CoRR*, abs/1412.6980.
- [18] Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123-140.