

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/358030621>

Cassava Leaf Disease Recognition Using Convolutional Neural Networks

Conference Paper · December 2021

DOI: 10.1109/ICOT54518.2021.9680655

CITATIONS

26

READS

1,394

2 authors:



Seksan Mathulapransan
Kasetsart University

20 PUBLICATIONS 147 CITATIONS

[SEE PROFILE](#)



Kitsana Lanthong
Kasetsart University

3 PUBLICATIONS 55 CITATIONS

[SEE PROFILE](#)

Cassava Leaf Disease Recognition Using Convolutional Neural Networks

Seksan Mathulapransan and Kitsana Lanthong

Department of Computer Engineering, Faculty of Engineering at Kamphaeng Saen
Kasetsart University

Nakhon Pathom, Thailand

seksan.m@ku.th, kitsana.lan@ku.th

Abstract—This study aims to investigate various kinds of convolutional neural networks to classify cassava leaf diseases. The objectives are that we want to survey the performance of current well-known CNN models including VGGs, ResNet, DenseNet, and Inception in term of classification performance and plan to extend the study of cassava disease recognition in the future. From our experiments, the best model is DenseNet121 with brightness augmentation, which achieves classification accuracy at 94.32% and F1-score at 92.13%.

Keywords—cassava, cassava disease recognition, plant disease recognition, convolutional neural network, deep learning

I. INTRODUCTION

Cassava is one of the economic plants in the world, which usually grows up in tropical lowland such as East, West, and Central Africa, South America, South Asia, Southeast Asia, etc. Cassava diseases are the main problem of farmers who want to control them to achieve good yield in term of both quality and quantity [1]. Usually, cassava farmers identify the type of diseases manually using their experience. Therefore, this way depends on each farmer's ability and contain high risk to loss the yield if there is an error from miss identifying.

Artificial intelligence (AI) models, especially machine learning and deep learning, are now commonly used in many research areas of data analysis, including the detection and recognition of plant diseases [2]–[5] because they have high performance and are suitable for many kinds of data including image, sound, text, number, etc. However, the key condition is that there must be appropriate and sufficient data for teaching the model.

Abdullakasim *et al.* [6] used image analysis technique with artificial neural network (ANN) to detect brown leaf spot (BLS) disease from leaf image. They used contrast indices extracted from red, green, and blue as well as hue, saturation, and intensity (HSI) as the feature and binary classified by ANN. Tuhaise *et al.* [7] studied a number of traditional classifiers including naïve Bayes, support vector machine (SVM), decision tree, nearest neighbors, and random forest to pixel-wised classify cassava brown streak disease (CBSD). They concluded that the strategy to use these algorithms to do automated classification system has a good potential and can be developed to be infield tool for the cassava farmer such as a mobile app.

For the use of deep learning, Ramcharan *et al.* [4] presented the detection of cassava diseases in Tanzania with Inception model based on photographic leaf data. The scope of this study includes five types of cassava diseases: brown leaf spot disease, red mite damage, green mite damage, cassava brown streak disease, and cassava mosaic disease.

The results are better than traditional machine learning models, which are KNN and SVM, and average accuracy is 93%. This team also extended the research by developing a mobile based application using a convolutional neural network (CNN) [8]. Sambasivam *et al.* [5] studied the detection of cassava diseases from leaf images in Uganda using convolutional neural networks. The dataset used in this study is imbalanced including five classes: cassava mosaic disease, cassava brown streak disease, green mite damage, cassava bacterial blight, and healthy with 4445, 2808, 1287, 1144 and 316 images, respectively. This research proposed a method to use class weight, focal loss, and synthetic minority over-sampling (SMOS) to address this imbalanced dataset. The proposed model was able to predict cassava diseases with an average accuracy of 93%.

This study aims to investigate a set of convolutional neural networks to classify cassava leaf diseases. The objectives are that we want to survey the performance of current well-known CNN models including VGGs, ResNet, DenseNet, and Inception in term of classification accuracy and plan to extend the study of cassava disease recognition as our future works.

II. CONVOLUTIONAL NEURAL NETWORKS

In deep learning, convolutional neural network is a deep neural network that has at least one convolutional layer inside its architecture. Usually, visual imagery is the data type that is mostly applied to CNN. In the convolutional layer, the main operation is the matrix convolution of the filters (or kernels), which usually have different sizes in different layers, and then the result of each single convolution is used as a shared weight to create a new node in the next layer [9]. When construct the architecture and assign its parameters properly, CNN can extract the contained information of the input image and classify the classes very well. An example of CNN architecture is shown in Fig. 1.

Nowadays, CNNs is developed to be more complex architecture with the purpose of solving more difficult and variant problems [10], [11], such as shortcut-connection network and parallel convolution network. The shortcut connection network is a special kind of CNN, which has shortcut paths of their weights to the next block inside the architecture. This is a way to create more hidden layers of the network without the gradient vanishing problem since some weights are bypass directly to the next layer without processing. The well-known models in this group are ResNet [12] and DenseNet [13]. In ResNet, the model is constructed by connecting several blocks of convolution layers, called residual block. The use of bypass is applied to transfer weights from a group of convolution layers to the next group, as the weights are skipped for some convolutions. In DenseNet, the

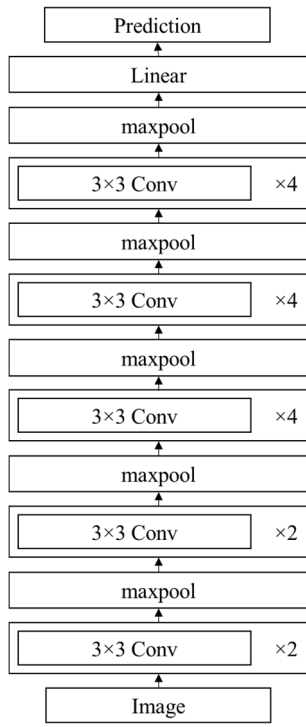


Fig. 1. An example of CNN architecture (VGG19 [14]).

model is constructed by a group of dense blocks. Each block is connected to a single convolution layer with a pooling layer before the next block. Inside the dense block, the inputs of each layer are collected from several bypasses connected from

all their previous layers within the block. On the other hand, the output is simply sent to each subsequent layer.

For the parallel convolution neural network, it does not try to stack more hidden layers to make the network go deeper but wider by using the parallel convolution layers with different filter sizes. Inception [15] is an example of this type, which was designed to solve the variation of focused object, e.g., size, position, angle, etc. Since it is hard to select the proper filter size of convolution layer when there is huge variation especially the object size. The smaller size of filter is good for more local while the bigger size is good for more global. Therefore, the different kernel sizes operating parallelly and simultaneously is a superb solution, which can help the accuracy as well as the computational cost.

III. CASSAVA LEAF DISEASE DATASET

The dataset used in this study is from the competition of cassava leaf disease classification in Kaggle [16]. This competition is a section of the fine-grained visual-categorization workshop at CVPR 2019. The dataset contains five classes including four disease classes and healthy class. The goal is to learn a model to classify a given image into these five categories. These images were collected in Uganda by local farmers taking them from their gardens. Then, the images were labelled by the plant pathologist from the National Crops Resources Research Institute (NaCRRI) in collaboration with the AI laboratory in Makerere University, Kampala.

The four diseases in this dataset include cassava bacterial blight (CBB), cassava brown streak disease (CBSD), cassava

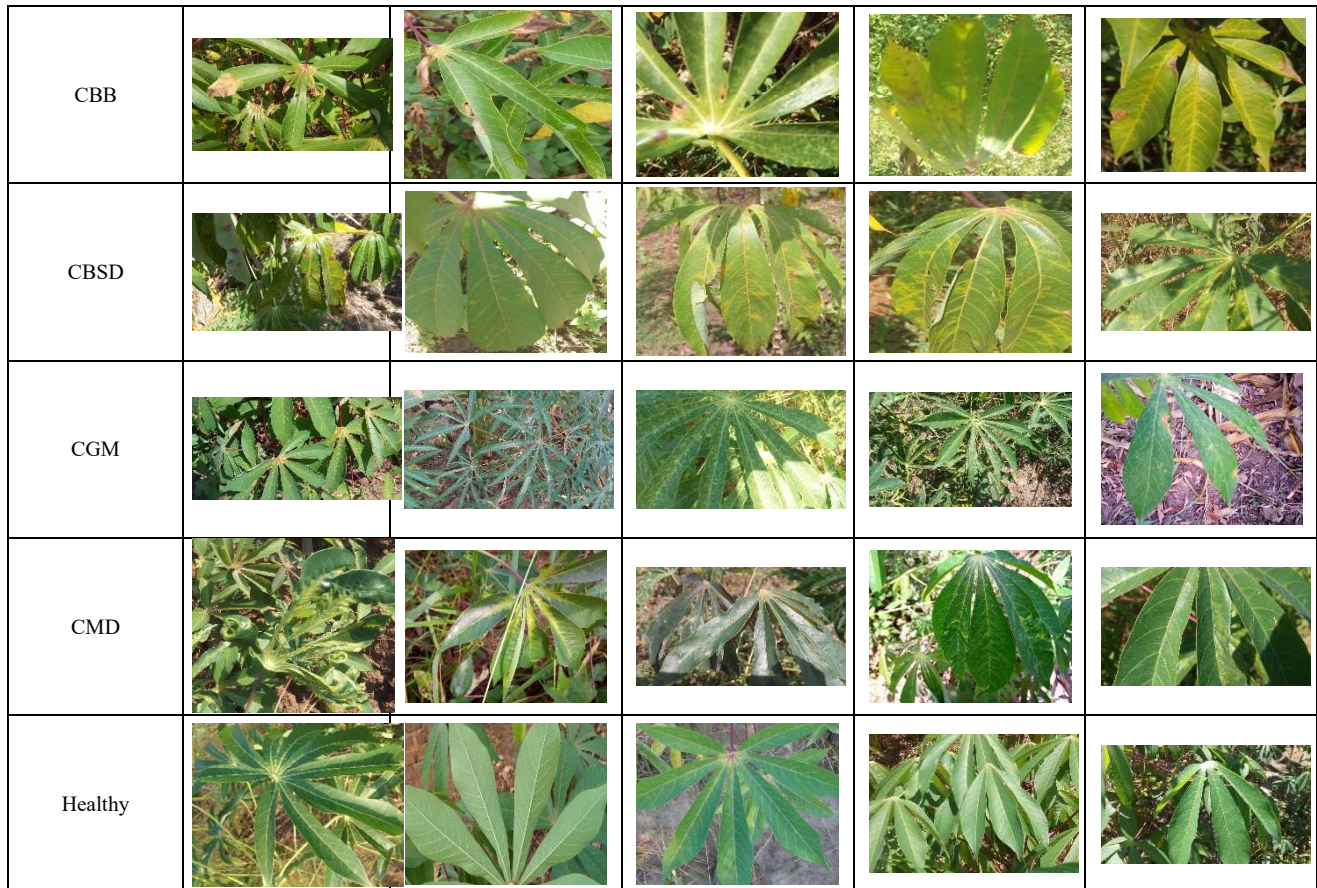


Fig. 2. Examples of cassava leaf disease dataset [16], which has 5 classes including cassava bacterial blight (CBB), cassava brown streak disease (CBSD), cassava green mite (CGM), cassava mosaic disease (CMD), and healthy.

TABLE I. CASSAVA LEAF DISEASE DATASET USED IN THIS STUDY

Class	Number of images
CBB	466
CBSD	1443
CGM	773
CMD	2658
Healthy	316
Total	5656

green mite (CGM), and cassava mosaic disease (CMD). The dataset contains more than 22,031 cassava leaf images. However, since the competition is not completed yet, the labels for test images are not revealed. Therefore, we use only the images from training set, which has 5656 cassava leaf images. The conclusion of class number and the examples of each class images are shown in TABLE I. and Fig. 2, respectively.

IV. EXPERIMENTAL RESULTS

The performance of the selective models of deep neural networks to classify cassava leaf diseases is revealed through a set of experiments in this section. Firstly, various kinds of CNN models were investigated. After that, five kinds of basic augmentations were applied to the best performing model from the first experiment. The results are reported in term of classification accuracy, confusion matrix, and F1 score. The classification accuracy can be defined as follows.

$$Accuracy = \frac{N_{correct}}{N_{total}} \quad (1)$$

where $N_{correct}$ represents the number of correctly classified samples, N is the number of total test samples.

Confusion matrix is used to express the classification performance in term of both quantity and quality. Usually, it is formed as a table representing the score of classifying the test samples. In the table, each row represents the counted samples (or percentage) of a predicted class while each column represents the counted samples (or percentage) of an actual class (or vice versa). This means we can see the wrong prediction clearly by using this evaluation because it show the label of miss classifying. The last evaluation method is F1 score, which is used to measure the evaluated model by considering not only true positive but also false positive and false negative. F1 score is calculated as follows.

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \quad (2)$$

$$Recall = \frac{N_{TP}}{N_{TP} + N_{FN}} \quad (3)$$

and

$$F1 \text{ score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

where N_{TP} , N_{FP} , and N_{FN} represent the number of true positive, false positive, and false negative, respectively.

The experimental setting used to do the designed experiments is fixed as follows. To make the reliability of the following results, 5-fold cross validation was carried out to evaluate the dataset. According to the network structure, all

images were resized to 224×224 pixels for feeding to the input layer. The batch size was set to 32, with the epoch equals to 10. The learning rate for all models was set to 0.0001. The experiments were run on a desktop computer with the GPU: GeForce RTX 2060 SUPER, compute 1.7, having 2176 CUDA cores, 8GB GDDR6 VRAM, CPU: core i3-9100F 3.6 GHz (4 cores, 4 threads), RAM: 8 GB available, and Disk: SSD SATA3 240 GB.

A. Comparison of different models

In this experiment, various CNN models were investigated including the basic CNN (see Fig. 3), VGG16 [14], VGG19 [14], ResNet50 (the 2nd version) [17], DenseNet121 [13], and Inception (the 3rd version) [18] were performed cassava leaf disease recognition. The results in term of classification accuracy are shown in TABLE II.

The results in TABLE II. present that the best model performed in this experiment is DenseNet121, which achieves classification accuracy at 80.52%. Since this score is quite far from the second place, which is Inception with 77.25% of classification accuracy. Therefore, we selected only DenseNet121 to combine with many methods of augmentations to find the best combination in the next experiment. Moreover, TABLE III. presents the confusion matrix of DenseNet121, which exhibits that the most accurately classified class is CMD and the worst classified class is CBB since this class has a huge intersection with CBSD.

B. Comparison of augmentation methods with the selected model

This section illustrates various methods of basic augmentations to search for the proper way to handily enhance the classification performance of the best classifier from the previous experiment, which is DenseNet121. Since this study focuses on the classifiers, which are CNN based models, the augmentations used are only the basic operations, which are horizontal flip, vertical flip, rotation, brightness adjustment, and zoom adjustment.

```
A basic CNN
Sequential()
Layer1: Conv2D(64, (5, 5), activation='relu',
input_shape=(224, 224, 3))
Layer2: MaxPooling2D(2, 2)
Layer3: Conv2D(128, (5, 5), activation='relu')
Layer4: MaxPooling2D(2, 2)
Layer5: Conv2D(128, (5, 5), activation='relu')
Layer6: Flatten()
Layer7: Dense(128, activation='relu')
Layer8: Dense(number_of_classes, activation='relu')
```

Fig. 3. Configuration of a basic CNN.

TABLE II. CLASSIFICATION ACCURACY OF COMPARED MODELS

Model	Accuracy
CNN	60.18
VGG16	70.26
VGG19	71.44
ResNet50	75.76
DenseNet121	80.52
Inception	77.25

TABLE III. CONFUSION MATRIX OF DENSENET121

True class	CBB	43.99	34.98	6.22	8.58	6.22
	CBSD	5.61	82.95	2.56	6.38	2.49
	CGM	3.62	6.47	69.86	16.17	3.88
	CMD	1.58	3.72	3.42	89.99	1.28
	healthy	4.11	13.61	4.43	8.23	53.48
		CBB	CBSD	CGM	CMD	healthy
Predicted class						

TABLE IV. CLASSIFICATION ACCURACY OF DENSENET121 WITH COMPARED AUGMENTATIONS

Augmentation method	Accuracy
Horizontal flip	86.47
Vertical flip	83.61
Rotation	80.55
Brightness adjustment	94.32
Zoom adjustment	85.47

TABLE V. CONFUSION MATRIX OF DENSENET121 WITH BRIGHTNESS ADJUSTMENT

True class	CBB	86.37	7.62	1.93	2.68	1.39
	CBSD	2.29	94.46	0.73	1.73	0.80
	CGM	2.07	2.13	88.87	6.14	0.78
	CMD	0.47	0.92	0.81	97.40	0.40
	healthy	1.11	3.32	0.95	1.74	92.88
		CBB	CBSD	CGM	CMD	healthy
Predicted class						

TABLE VI. F1 SCORES OF EACH DISEASE CLASS OF DENSENET121 WITH BRIGHTNESS ADJUSTMENT

Disease	Precision	Recall	F1-score
CBB	87.45	85.94	86.69
CBSD	94.35	94.25	94.30
CGM	94.20	88.23	91.12
CMD	96.76	97.18	96.97
Healthy	90.43	92.72	91.56
Average	92.638	91.66	92.13

The results of all augmentation methods with DenseNet121 are shown in TABLE IV. From the results above, the brightness adjustment is the best augmentation to improve the classification performance of DenseNet121, which achieve classification accuracy at 94.32%. TABLE V. presents the confusion matrix of this combination. From this matrix, the most accurately classified class is CMD and the worst classified class is CBB, which their classification accuracies are 97.40 and 86.37, respectively. Same reason as the previous experiment, CBB has a huge intersection with CBSD. However, with brightness augmentation, this intersection is decreased from 34.98% to 7.62%. Therefore, the mean accuracy is significantly increased from 80.52% to 94.32%. Moreover, to investigate deeper, precision, recall, and F1 score of each class and their average performed by DenseNet with brightness augmentation are presented in TABLE VI.

V. CONCLUSION

This study focuses on exploring various kinds of CNN-based models to classify cassava leaf diseases. The selected models including basic CNN (manual configuration), VGGs,

ResNet, DenseNet, and Inception were compared in term of classification performance. Furthermore, a number of augmentation methods including horizontal flip, vertical flip, rotation, brightness adjustment, and zoom adjustment were applied with the best-performed classifier. The experimental results show that, without augmentation, DenseNet121 achieve highest classification score, which is 80.52%. When applied with brightness adjustment augmentation, its classification accuracy is increased to 94.32%

Future works of our team can be divided into two folds. First, we plan to study this problem in Thailand and create a new dataset from local cassava farms. Second, the models to study should be extended to cover more kinds of deep learning as well as model improvement to enhance the classification performance.

REFERENCES

- [1] J. P. Legg *et al.*, "Chapter Four - Cassava Virus Diseases: Biology, Epidemiology, and Management," in *Control of Plant Virus Diseases*, vol. 91, G. Loebeinstein and N. I. Katis, Eds. Academic Press, 2015, pp. 85–142.
- [2] S. Mathulapransan, S. Patarapuwadol, K. Lanthong, D. Jetpipattanapong, and S. Sateanpattanakul, "Rice Disease Recognition Using Effective Deep Neural Networks," *J. web Eng.*, vol. 20, no. 3, pp. 853–878, 2021.
- [3] D. I. Patricio and R. Rieder, "Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review," *Comput. Electron. Agric.*, vol. 153, no. April, pp. 69–81, 2018, doi: 10.1016/j.compag.2018.08.001.
- [4] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, "Deep learning for image-based cassava disease detection," *Front. Plant Sci.*, vol. 8, no. October, pp. 1–7, 2017, doi: 10.3389/fpls.2017.01852.
- [5] G. Sambasivam and G. D. Opiyo, "A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks," *Egypt. Informatics J.*, vol. 22, no. 1, pp. 27–34, 2021, doi: 10.1016/j.eij.2020.02.007.
- [6] W. Abdullakassim, K. Powbunthorn, and J. Unartngam, "An Images Analysis Technique for Recognition of Brown Leaf Spot Disease in Cassava," *Tarim Makinalari Bilim. Derg.*, vol. 7, no. 2, pp. 165–169, 2011.
- [7] J. Tuhaise, J. A. Quinn, and E. Mwebaze, "Pixel Classification Methods for Automatic Symptom Measurement of Cassava Brown Streak Disease," *Proc. 1st Int. Conf. Use Mob. ICT Africa*, no. December, pp. 6–10, 2014, [Online]. Available: <https://github.com/tjovia/CBSD.git>.
- [8] A. Ramcharan *et al.*, "A mobile-based deep learning model for cassava disease diagnosis," *Front. Plant Sci.*, vol. 10, no. March, pp. 1–8, 2019, doi: 10.3389/fpls.2019.00272.
- [9] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks," pp. 1–11, 2015, [Online]. Available: <http://arxiv.org/abs/1511.08458>.
- [10] W. Rawat and Z. Wang, "Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, 2017, doi: 10.1162/neco_a_00990.
- [11] A. Dhillon and G. K. Verma, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Prog. Artif. Intell.*, vol. 9, no. 2, pp. 85–112, 2020, doi: 10.1007/s13748-019-00203-0.
- [12] K. He and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017, doi: 10.1109/CVPR.2017.243.
- [14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

- [15] C. Szegedy *et al.*, “Going Deeper With Convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9.
- [16] E. Mwebaze, T. Gebru, A. Frome, S. Nsumba, and J. Tsubira, “iCassava 2019 Fine-Grained Visual Categorization Challenge.” 2019.
- [17] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.
- [18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 2818–2826, 2016, doi: 10.1109/CVPR.2016.308.