Classifying barako coffee leaf diseases using deep convolutional models

Francis Jesmar Perez Montalbo [a,b,1,*], Alexander Arsenio Hernandez [a,2]

[a] Technological Institute of the Philippines, 363 Pascual Casal St, Quiapo, Manila, Metro Manila, 1001, Philippines
[b] Batangas State University, Rizal Ave, Extension, Batangas City, 4200, Philippine
[1] mfjpmontalbo@tip.edu.ph; [2] alexander.hernandez@tip.edu.ph
* corresponding author

ARTICLE INFO

ABSTRACT

This work presents the application of recent Deep Convolutional Models (DCM) to classify Barako leaf diseases. Several selected DCMs performed image classification tasks using Transfer Learning and Fine-Tuning, together with data preprocessing and augmentation. The collected dataset used totals to 4,667. Each labeled into four different classes, which included Coffee Leaf Rust (CLR), Cercospora Leaf Spots (CLS), Sooty Molds (SM), and Healthy Leaves (HL). The DCMs were trained using the partial 4,023 images and validated with the remaining 644. The classification results of the trained models VGG16, Xception, and ResNetV2-152 attained overall accuracies of 97%, 95%, and 91%, respectively. By comparing in terms of True Positive Rate (TPR), we found that Xception has the highest number of correct classifications of CLR, VGG16 with SM, and CLS, while ResNetV2-152 with the lowest TPR for CLR. The evaluated results indicate that the use of Deep Convolutional Models with an adequate amount of data, proper fine-tuning, preprocessing, and transfer learning can yield efficient classifiers for identifying several Barako leaf diseases. This work primarily contributes to the growing field of deep learning, specifically for helping farmers improve their diagnostic process by providing a solution that can automatically classify Barako leaf diseases.

1. Introduction

The coffee industry made immense global contributions to society, placing as the second most traded commodity next to crude oil worldwide. With an estimated amount of 15 billion trees planted, it supports the demand of 25 million producers around several countries [1]. In most Asian territories, the coffee industry paves employment for several families to cope up with their daily needs. In the Philippines, the Coffea Liberica is a popular coffee variant referred to as Barako. The sought-after product possesses a distinct flavor and aroma that interest most consumers. Unlike other varieties, the Barako tree is difficult to grow as it consumes a larger land area, making it a less encouraging option for farmers. Also, Barako cultivation is greatly affected by widespread diseases. According to the Philippines coffee industry roadmap, in 2015, Liberica only yielded 257 metric tons (MT) of coffee beans, contributing only 1% to the whole coffee production. At the same time, Robusta produced an average of 24,924 MT, providing 69%, followed by Arabica with 8717 MT at 24%, and Excelsa with 2273 MT at 6%. Since the rust invasion of 1896, Barako became less enticing to grow due to farmers opting for other alternatives. Considering that Excelsa being less vulnerable against drought and most infections [2][3].

Even today, it remains a challenge for experts to provide an immediate diagnosis for plant diseases. The lengthy procedure frequently turns to a massive spread of infection that causes excessive losses [4].

Moreover, such conditions originated from various types of fungi. These pathogens found on the leaves are highly contagious and spreads rapidly if not given immediate attention. Indicated in a study, that roughly 10% of the plant economy worldwide is being affected due to the devastating effects of plant infections and infestations [5]. Plant disease diagnosis involves complicated procedures like symptom analysis, pattern recognition, and several forms of tests on leaves that consume extensive periods and resources [6]. In most cases, improper diagnosis can cause immunity or less susceptibility of plants to treatment. The complexity of plant disease diagnosis caused farmers to produce less quantity and quality yield [7].

With the advancement of technology, researchers discovered an alternative method to preserve natural resources. The continuous desire for improving Artificial Intelligence has primarily contributed to the increased performance of Deep Learning (DL), a trendsetting technology that can deliver innovative solutions for future endeavors [8]. In agriculture, DL aims to surpass existing human capabilities as it provides a rapid, accurate, and less costly approach to diagnose plant disease [9][10].

The growing interest of DL in agriculture leads to various studies that proved visual assessment is exceptionally reliable for plant disease diagnosis. In the last years, researchers have been producing DL solutions for agriculture in terms of classifying diseases and species using Convolutional Neural Networks (CNN) [11][12]. A DL model like CNN is composed of convolution layers that convolve a 3x3 filter over an image to generate feature or activation maps. The subsequent activations then pass through a set of down-sampling layers that reduce its values in half. The CNN then classify based on the activations' probability, using a SoftMax function [13].

Recent studies had applied CNN models to classify leaf diseases. Marcos *et al.* [14] devised a CNN model with lesser depth and complexity compared to a more advanced deep convolutional model (DCM). Their work attained an accuracy score of 95% during the span of 500 epochs, with only a 0.10 loss using only 159 coffee leaf images. The results indicate that CNN has high potentials in contributing to plant disease diagnosis. However, the study added that using an advanced CNN model could improve disease diagnosis. In another study, Esgario *et al.* [15] trained several Deep Convolutional Models (DCM), including ResNet50, and VGG16 using 1747 Arabica leaf images. The researchers classified the biotic stress and its severity level. The trained DCNNs attained 95.47% with VGG16, which determined several biotic diseases, while ResNet50 efficiently validated each leaf condition with a 95.63% accuracy rate. However, when the models only performed the classification of symptoms, the accuracy went up to 97%. In their conclusion, increasing the volume of the dataset for DL models can further enhance the classification effectiveness. Employing image processing, relevant features of coffee diseases like coffee leaf rust (CLR), leaf miner, CLS, bacterial blight, brown leaf spots, and blisters can be isolated. A developed system by Barbedo [16] automatically eliminated any irrelevant features, like the background and unaffected areas, to reduce diagnosis errors and increase the accuracy to classify diseases apart.

Furthermore, Bergstra and Bengio [17] indicated that generating an efficient DL model for classification, proper tuning of hyper-parameters is imperative. Training without prior adjustment of hyper-parameters can lead to poor performance, as to models that were tuned. Bergstra and Bengio [17] also added that hyper-parameters act like the "bells and whistles" in a learning algorithm, that shift the weights to generate the lowest possible errors.

In this work, we trained several DCMs to classify Barako leaf diseases for immediate, inexpensive, and accurate results. It is worth mentioning that in our work, only limited types of leaf diseases are present due to some restrictions declared by farm owners in collecting samples. However, we still guarantee a substantial contribution to improve the cultivation of Barako coffee. Included also are future works determined for further enhancements of this work.

2. Method

In this section, we provide the following DCMs trained for the Barako leaf disease classification, which consists of the following: Xception, ResNetV2-152, and VGG16 [18]. The given description of

each DCM provides a further understanding of how each works accordingly. Our method also employs the use of data preprocessing, data augmentation, transfer learning, and fine-tuning.

2.1. VGG16

With the improvements in computing power, deep-layered networks became possible to train. VGG emanated with several configurations. However, the 16-depth version achieved better results than its counterparts in terms of application [19].

VGG focuses specifically on the method of stacking more layers to improve classification accuracy. Fig. 1 illustrates VGG16's architecture. VGG16 accepts an input of 224x224x3 and uses a 3x3 convolving filter for all color channels simultaneously. The convolution process then generates a dot product output called an activation map or filter. The activation map helps the classifier to recognize images on the fully connected (FC) layers to calculate results. The architecture has a series of 2x2 Max-Pooling (MP) layers with a stride of two that down-sizes the image values in half before passing to the Softmax classifier. The classifier includes hidden units of 4096 neurons equipped with a ReLU activation. ReLU reduces pixel values to zero or 255 in a non-linear way and proved to increase model efficiency [20][21].
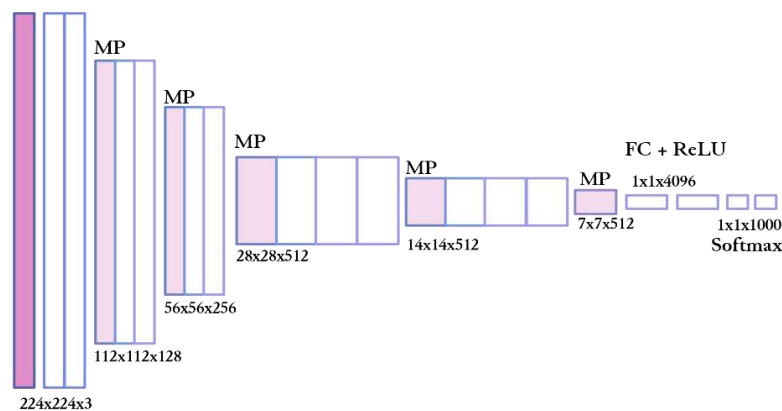

Fig. 1. VGG16 Architecture

2.2. Xception

Based on the study of Szegedy *et al.* [22], depth-wise separable convolutions outperformed the previous InceptionV3 module with a smaller number of parameters. However, the study indicated that parameter count did not contribute mainly to the improvements, but rather, on how Xception used it. The new profound method of reversing the process of the standard inception module, made Xception stand out with lesser computations and complexity [23]. Such findings interested us in applying Xception in this work. For a better interpretation, we provided Fig. 2 that illustrates the procedure of Xception.

The method begins by using a 1x1 pointwise convolution on a spatial image that iterates over every color channel, followed by a depth-wise convolution that convolves using a channel-wise filter for each color channel individually. The individual channels then concatenate to form a new set of spatial filters with lighter computations. This method indicates the contrast in the extraction process of Xception with VGG16, as it convolves all color channels at the same time, with a larger kernel size [22].
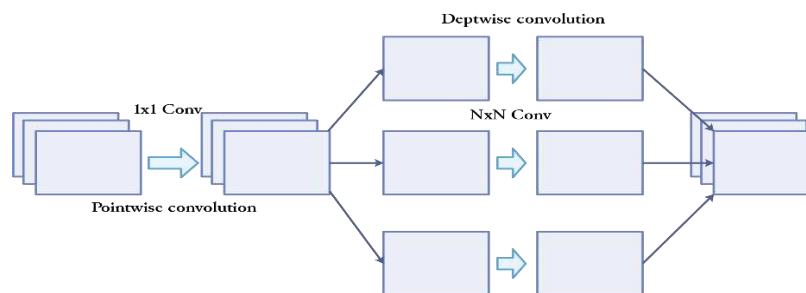

Fig. 2. Xception Block

2.3. ResNet V2

Previous works stated that stacking more layers on the network can increase effectiveness. However, He *et al.* [24] indicated that a deeper stack of layers could saturate the model accuracy over time. Training deeper models can lead gradients to vanish or explode, resulting in lower accuracy rates. The development of residual blocks solved the problem of deeper models using a skip or shortcut connection. ResNetV2, a later version of the original ResNetV1, proposed the use of identity mappings as shortcuts. With this, the V2 method now propagates a signal through the skip connections to earlier blocks in the network, unlike V1. Fig. 3 illustrates the re-arrangement of the following layers of the original ResNet V1 block compared to the V2 [25][26].
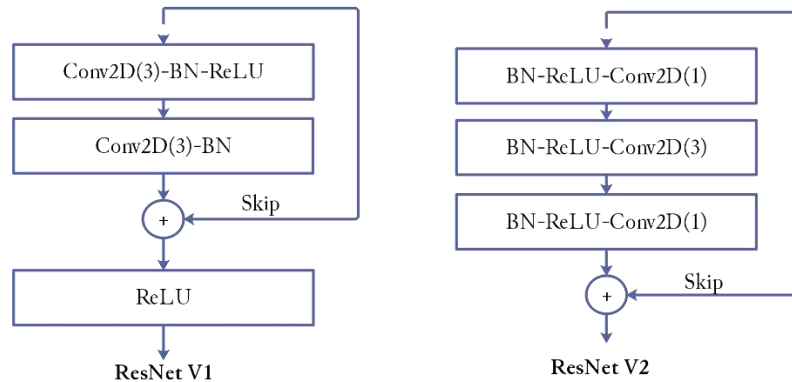
Fig. 3. ResNet V1 and V2 Blocks

2.4. Data Preparation and Specifications

In this work, we considered three major leaf diseases of Barako: Coffee Leaf Rust (CLR), Cercospora Leaf Spots (CLS), and Sooty Molds (SM). We also included images of Healthy Leaves (HL) to prevent any false diagnosis from the rest. The collected samples came from a local coffee farm, taken in a controlled environment using a smartphone with a shooting resolution of 3247x3247 pixels. Together with the presence of proper lighting and setup, preprocessing became less challenging. According to some works, poorly captured images caused by insufficient light, blurriness, and other forms of noise could generate lower classification performance [27]-[29]. Hence, we guaranteed to avoid any occurrence of shadow, background noise, and loss of pixel value that may affect the learning and classification process as much as possible.

In Fig. 4, we present our samples captured with poor (a) and improved (b) lighting. In Fig.4(a), the immense presence of shadows cast over the subject due to poor lighting conditions. With the poorly captured image, we decided to discard it from our training sample to prevent noisy data inclusions. Instead, we decided to retake another with additional lighting to reduce the chances of reproducing shadows in (b). Also, to further improve the quality of our samples, we removed the background of each sample to avoid unnecessary particles, shadows, or any irrelevant features that the model might pick up during training shown in (c).
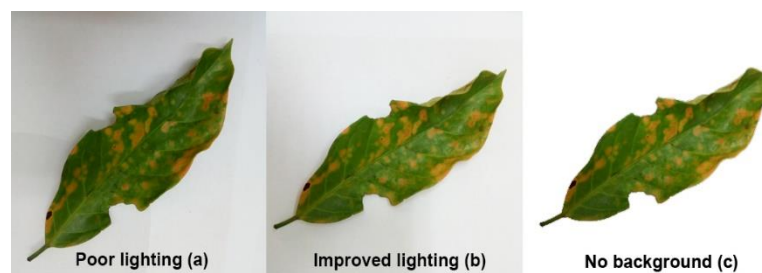

Fig. 4. Image Samples with Poor lighting (a), Good lighting (b), and No Background (c)

Likewise, an appropriate label for each image is necessary for the experiment. The leaves collected reached up to 4,667 and were annotated appropriately by an identified specialist, as presented in Table 1. Improper labeling can cause severe problems if not done correctly [30]. This work prevented such circumstances from happening.

For added efficiency, we applied a minimal image processing to resize the input sizes in terms of height, width, and depth to generate a standardized dimension of our data entering each model. For VGG16 and ResNetV2-152, we used a 224x224x3 while, a 299x299x3 shape dimension for Xception. The reason is to utilize the most effective pre-defined sizes given by the authors of each model. However, reducing the required maximum dimensions can limit the extraction of better features that can degrade the model's effectiveness to classify [31]. On the other hand, enlarging the dimensions would only consume higher computational cost with minimal to no improvements, as the authors provided fixed measurements that are already optimal for each architecture [21][23][25]. To prevent inconsistency, noisy data, expensive training costs, and poor accuracy, we applied these preprocessing methods to improve the overall performance of all models [32].

Table 1. Dataset and Specifications

Class	Train Quantity	Validation Quantity	Sample Image	Leaf Pattern	Description
HL	794	136			Bright and green without any forms of damage or stain
SM	1683	196			Contains dark molds with a bit of curling
CLR	752	116			Has halo-like yellowish to brownish spots
CLS	794	196			Covered with brownish dried spots
Total	**4023**	**644**			

2.5. Data Augmentation

Training a DL model with low quantities of data can result in poor performance [33]. Even with our collected number of images, it is still insufficient for a useful DL model to work [34]. To cope up with this problem, we decided to augment our data. We used the image data generator instance from the Keras API to perform an automatic augmentation of images during the training process [35]. With this, passing train samples in each model during training will automatically augment the images without the need for laborious manual transformations.

With the numerous ways of augmentation process, we only considered the standard techniques like zoom, shear, rotation, height, and width shift, horizontal and vertical flips, as it can increase learning patterns for the model without too much alteration shown in Fig. 5(a). Needless augmentation can cause the model to misclassify due to heavy distortion or skew the images too much, as shown in Fig. 5(b). Features like colors and location of disease patterns may disappear or become unrecognizable [36]. Therefore, we only applied techniques that can increase the training volume while avoiding such problems.

Fig. 5 illustrates the set of good and bad augmented samples. Images in (a), are the only ones we considered to apply, while (b) resembles a heavily distorted sample that can cause problems during the classification task. Moreover, data augmentation was only performed on the training data to prevent the chances of bias and incidence of data leakage [37][38].
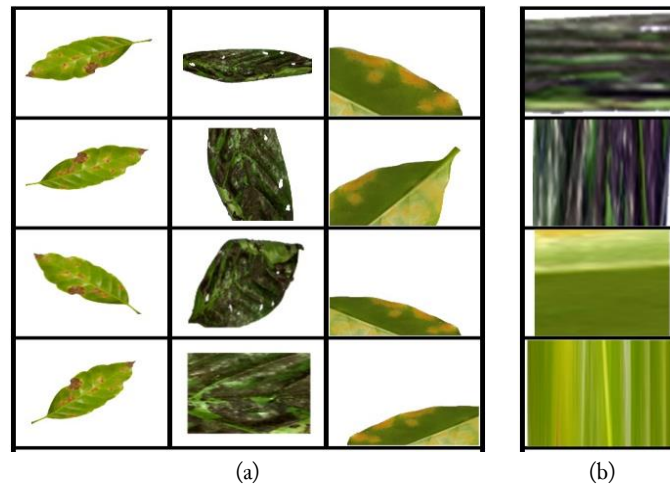
(a) (b)

Fig. 5. Samples with Acceptable (a) and Poor (b) Augmentation

2.6. Hyper-parameters

Before the initial training commences, we considered a domain of hyper-parameters. The following values in Table 2 originated from a review and stochastic selection of several hyper-parameters from previous works [39]. Our indicated combinations of values were tuned to yield the highest possible accuracy for all three models.

Table 2 presents our hyper-parameter settings selected based on a survey study [40]. To train our models, we used Stochastic Gradient Descent (SGD) as our optimizer tuned based on momentum, batch size, and hidden units according to each model's characteristics. The slower learning phase of SGD improved the model performance to classify with less overfitting than a faster learning optimizer like Adam and RMSProp. According to Keskar and Socher [41], with extended training periods, SGD can still outperform both RMSProp and Adam. In which, we performed the training of our models in a span of 100 epochs.

Table 2. Hyper-Parameter Settings

Model	Hyper-Parameter Values					
	Optimizer	*LR*	*Momentum*	*Epoch*	*Batch Size*	*Hidden Unit*
Xception	SGD	0.1^{e-3}	0	100	64	256
ResNet V2-152	SGD	0.1^{e-3}	0.5	100	32	256
VGG16	SGD	0.1^{e-3}	0.9	100	32	128

2.7. Transfer Learning and Fine-Tuning

The following DCMs selected, Xception, ResNetV2-152, and VGG16, trained using our collected 4023 train data. However, generating feature parameters from a low volume of data could lead to an inferior learning process and accuracy [42].

Hence, to achieve better accuracy and a robust set of parameters, we transmitted the ImageNet weights from each pre-trained DCM to our models using Transfer Learning. ImageNet improved our trained model's capability to detect images based on edges, blobs, corners, and other essential feature parameters needed for image classification. This approach reduced the need for high-end resources during training compared to an extensive initialization of the entire weights from scratch [43].

However, the parameters of the pre-trained models originally trained to classify 1000 different classes like cars, planes, dogs, cats, and other unrelated inclusions using 1000 FC neurons instead of 4. Also, the original FC neurons of the pre-trained models from ImageNet did not contain any Barako diseases [44]. Having inappropriate FC neurons for the given task can lead to inaccurate or useless results [45].

To resolve this problem, we applied Fine-Tuning. With this approach, we crafted a new FC head consisting of only 4 FC neurons to classify our 4-classes. The new FC head replaced the previous 1000 FC Neurons of the pre-trained DCMs used to classify 1000-classes. Through this approach, we managed to tailor-fit the models and train with our new FC head containing a suitable number of FC neurons that included additional weights initialized only for HL, SM, CLR, and CLS while preserving the essential parameters from ImageNet.

3. Results and Discussions

To generate our results, we used a machine with an i5 Intel CPU running at 3.50GHz, an NVIDIA GeForce GTX 1070 GPU with 8.0GB of VRAM, and a 16.0 GB RAM indicated in Table 3. Our software tools involve the use of the Tensorflow framework and Keras API applications on a Jupyter Notebook. Table 3 presents the hardware specifications of our machine used for training.

Table 3. Machine Specifications

Component	Specification
CPU	I5-4690 @ 3.50GHz
GPU	NVIDIA GTX 1070 with 4095MB Memory
Memory	16.0GB DDR4-2133

3.1. Training Accuracy and Loss

During the training period of 100 epochs, we evaluated the models based on the growth of accuracy rates and the decrease of loss rates using cross-entropy as a multi-class loss function [46]. The cross-entropy loss function calculates errors between the actual training data to the model's prediction. A broader difference between the train to validation accuracy or loss indicates that the model experienced overfitting or underfitting that can affect the classification process. However, the closer the train to validation values show a convergence that results in a well-performing model [47][48].

Fig. 6 presents the training results of all three models. ResNetV2-152 (b) started with the lowest accuracy of 32% at the first epoch, followed by VGG16 (c) at 51%, and Xception at 60% (a). However, after the 100th epoch, VGG16 had the closest convergence in all three, followed by Xception, and ResNetV2-152. The plot presented was attained using our selected hyper-parameters. The model fit for accuracy was satisfactory without the occurrence of severe overfitting. The train and validation accuracy at the last epoch only had a difference of 4% for Xception, 6% for ResNetV2-152, and only 2% for VGG16, which indicated the best fit. However, basing only on accuracy rates does not entirely determine the effectiveness of a model to generalize. Therefore, we also evaluated each based on its loss per epoch to assess overall efficiency.
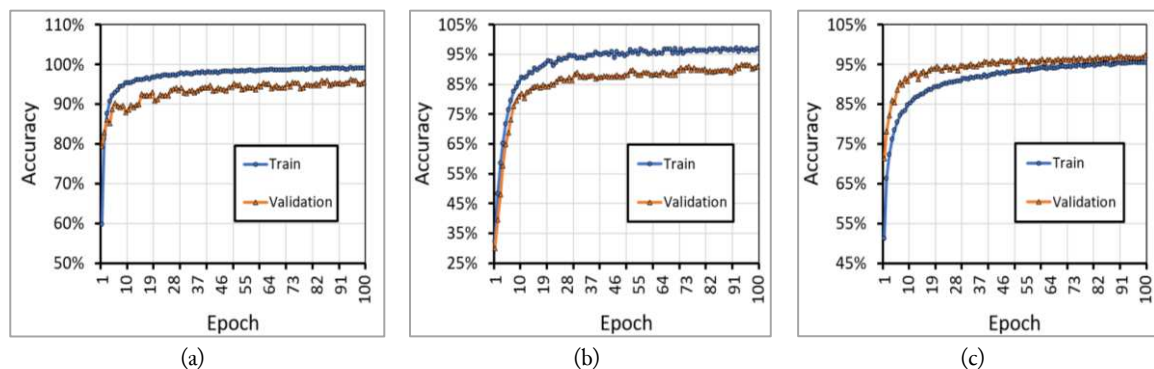

Fig. 6. Training Accuracy per Epoch: Xception (a), ResNetV2-152 (b), VGG16 (c)

Fig. 7 illustrates the loss graph. All three models had a continuous decrease of loss during training until the 100th epoch. Cases of overfitting and underfitting occurred in an earlier portion due to the lack

of patterns supported by the validation data. The lack of data remains a limiting factor for developing highly efficient plant disease detection models for DCMs [49].

Results between the difference of train and validation loss at the final epoch ended with 0.10 or 10% for Fig. 7(a) Xception, 0.31 or 31% for (b) ResNetV2-152, and 0.07 or 7% for (c) VGG16. To determine the effect of these results, we further investigated using a confusion matrix to calculate the overall accuracy and the numbers of correctly classified samples of each model.
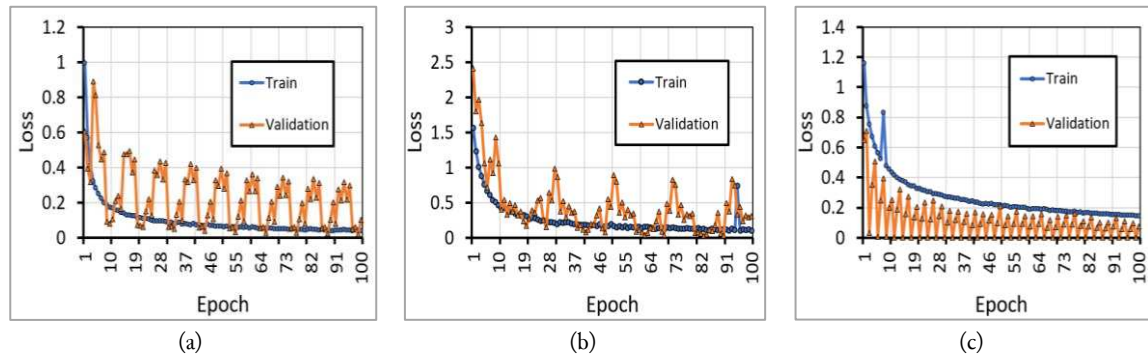
Fig. 7. Training Loss per Epoch: Xception (a), ResNetV2-152 (b), VGG16 (c)

3.2. Classification Performance

The diagnosis of Barako leaf diseases can suffer from misclassification due to their intricate patterns and similarities. With the confusion matrix, we can visually determine each model's performance on how it classified leaf diseases individually [50]. To validate our results, we used 644 validation samples to calculate the over-all classification accuracy. Fig. 8 illustrates the computed results of the three models after the training process using a confusion matrix. Each corresponding highlighted block on the diagonal trajectory indicates a correct diagnosis. Beyond the mentioned blocks are considered incorrect. Moreover, we also determined the percentage of correctly classified samples from each trained model with True Positive Rates (TPR). The TPR metric helps us to identify the number of correctly classified positive samples infected by diseases apart from the classified negative samples (HL). A model with a higher TPR percentage indicates a better classification performance of true positives [50].

Illustrated in Fig. 8(a), Xception attained 95.50% overall accuracy with its lowest TPR of 89.29% for SM, followed by 96.55% for CLR, and 98.50% for CLS while its highest TPR reached 99.26% for HL.

Fig. 8. Confusion Matrix: Xception (a), ResNetV2-152 (b), VGG16 (c)

ResNetV2-152 achieved an overall score of 90.83% (Fig. 8(b)). It has shown excellent performance with a 100% TPR for HL, 97.96% for CLS, and 84.69% for SM. However, it had difficulty in classifying CLR, resulting to a TPR of only 78.45%, making it the lowest among all TPRs. VGG16 (Fig. 8(c)) attained similar results with ResNetV2-152 in classifying HL at 100%, and its lowest TPR of 91.38%

for CLR. Other classes, like SM and CLS, reached TPRs of 96.94% and 98.98%, correspondingly. The overall accuracy of (Fig. 8(c)) was 97.20%, making it the dominant model followed by (Fig. 8(a)).

3.3. Discussions

In this section, we compared the results of other works that performed similar tasks of identifying diseases in coffee plants. It is worth mentioning that we cannot directly compare each study as we had different methods and data used. However, there are still comparable aspects in terms of objectives and the use of DCMs.

Table 4 presents the comparison of the top trained models of each work and its highest accuracy attained, as each author had multiple models. Our work had an accuracy of 97% in classifying four different leaf conditions (HL, SM, CLR, and CLS). Esgario et al. also had a score of 97%, which identified four biotic leaf stresses. Another work by Liang et al. focused on the severity of diseases and landed with 91%. Lastly, the work of Barbedo had achieved top accuracy of 88%. However, the work of Barbedo classified six different kinds of biotic stresses, making it the highest number of classes compared to all works presented.

Table 4. Comparison of Results from Other Works

Authors	Top Accuracy Attained
Our Work	**97%**
Esgario *et al.* [15]	**97%**
Marcos *et al.* [14]	95%
Liang *et al.* [51]	91%
Barbedo [16]	88%

The primary contribution of this work mainly lies in the data, collection method, and the processes performed. Among all works presented, this work considered the highest number of Coffea Liberica compared to others, which mainly used Arabica and Robusta leaves. The collection process included the use of proper lighting in a controlled environment added with preprocessing methods like background subtraction to remove any inappropriate image noise before training.

With the total collected dataset of 4,667, we selected to train and validate the classification performance of the newer architectures of DCMs, namely, Xception and ResNet152-V2 using Transfer Learning and Fine-Tuning. Unlike the other works that used previous versions of DCMs for their task. Added preprocessing and augmentation methods further improved this work to attain significant results to classify Barako leaf diseases.

4. Conclusion

This research classified Barako leaf diseases using notable recent DCMs to improve the process of diagnosis. We collected 4667 Barako leaf images from a local farm separated into a training dataset of 4023 and a validation set of 644. Each leaf sample is labeled into four classes by an expert with CLR, CLS, SM, and HL. To train our models, we applied transfer learning, fine-tuning, preprocessing methods, data augmentation, together with our selected hyper-parameters. VGG16, Xception, and ResNetV2-152 attained overall accuracies of 97%, 95%, and 91%, respectively. However, to our conclusion, the use of limited validation quantity led to heavy oscillations during validation. Nonetheless, each model still managed to attain significantly low error rates at the end of 100 epochs. Classification results based on TPR indicated that Xception could classify CLR samples better than the rest, while VGG16 tops with SM and CLS. At the same time, ResNetV2-152 had difficulty with CLR and attained the lowest performance in terms of TPR and overall accuracy. This work concludes that DCMs can potentially improve the diagnosis of Barako leaf diseases to help local farmers. Furthermore, this research can still scale to a more viable solution. We identified three paths for future researchers that may have an interest. First, to increase the data and patterns, which can highly contribute to improving classification performance. Second, to develop a specialized model that can identify unlearned patterns

beyond the given diseases without or low bias and variance. Last, we recommend having a practical application of the collected data and models for real-time use to diagnose and perform treatment even without farmers' physical presence.

Acknowledgment

Declarations

References

[1] M. Sanders, "Breeding for Coffee Leaf Rust Resilience in Coffea sp," *Natural Sciences Education*, vol. 48, no. 1, 2019, doi: 10.4195/nse2019.01.0102.

[2] "2017-2022 Philippine Coffee Industry Roadmap," *https://da.gov.ph*, 2017. [Online], available at: http://www.da.gov.ph/wp-content/uploads/2019/06/Philippine-Coffee-Industry-Roadmap-2017-2022.pdf. [Accessed: 5-Mar-2020].

[3] Bd, "Kapeng Barako Coffee Liqueur: Keeping The Spirit Of Batangas Coffee," *Business Diary Philippines*, Feb. 2019. [Online], available at: https://businessdiary.com.ph/16278/kapeng-barako-coffee-liqueur-keeping-the-spirit-of-batangas-coffee/. [Accessed: 17-Mar-2020].

[4] J. L. Badel and L. Zambolim, "Coffee bacterial diseases: a plethora of scientific opportunities," *Plant Pathology*, vol. 68, no. 3, pp. 411–425, Aug. 2019, doi: 10.1111/ppa.12966.

[5] R. Balodi, S. Bisht, A. Ghatak, and K. H. Rao, "Plant Disease Diagnosis: Technological Advancements and Challenges," *Indian Phytopathology*, vol. 70, no. 3, Jan. 2017, doi: 10.24838/ip.2017.v70.i3.72487.

[6] M. Riley, M. Williamson, and O. Maloy, "Plant Disease Diagnosis," *Plant Health Instructor*, 2002, doi: 10.1094/PHI-I-2002-1021-01.

[7] S. A. Miller, F. D. Beed, and C. L. Harmon, "Plant Disease Diagnostic Capabilities and Networks," *Annual Review of Phytopathology*, vol. 47, no. 1, pp. 15–38, 2009, doi: 10.1146/annurev-phyto-080508-081743.

[8] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[9] L. Yuan, Y. Huang, R. W. Loraamm, C. Nie, J. Wang, and J. Zhang, "Spectral analysis of winter wheat leaves for detection and differentiation of diseases and insects," *Field Crops Research*, vol. 156, pp. 199–207, 2014, doi: 10.1016/j.fcr.2013.11.012.

[10] A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar, "Machine Learning for High-Throughput Stress Phenotyping in Plants," *Trends in Plant Science*, vol. 21, no. 2, pp. 110–124, 2016, doi: 10.1016/j.tplants.2015.10.015.

[11] P. Jiang, Y. Chen, B. Liu, D. He and C. Liang, "Real-Time Detection of Apple Leaf Diseases Using Deep Learning Approach Based on Improved Convolutional Neural Networks," in *IEEE Access*, vol. 7, pp. 59069-59080, 2019, doi: 10.1109/ACCESS.2019.2914929.

[12] F. J. P. Montalbo and A. A. Hernandez, "An Optimized Classification Model for Coffea Liberica Disease using Deep Convolutional Neural Networks," 2020 16th IEEE International Colloquium on Signal Processing & Its Applications (CSPA), Langkawi, Malaysia, 2020, pp. 213-218, doi: 10.1109/CSPA48992.2020.9068683.

[13] Z. Zheng, S. Pan and Y. Zhang, "Fruit Tree Disease Recognition Based on Convolutional Neural Networks," *2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science*

and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Shenyang, China, 2019, pp. 118-122, doi: 10.1109/IUCC/DSCI/SmartCNS.2019.00048.

[14] A. P. Marcos, N. L. Silva Rodovalho and A. R. Backes, "Coffee Leaf Rust Detection Using Convolutional Neural Network," *2019 XV Workshop de Visão Computacional (WVC)*, São Bernardo do Campo, Brazil, 2019, pp. 38-42, doi: 10.1109/WVC.2019.8876931.

[15] J. G. Esgario, R. A. Krohling, and J. A. Ventura, "Deep learning for classification and severity estimation of coffee leaf biotic stress," *Computers and Electronics in Agriculture*, vol. 169, p. 105162, 2020, doi: 10.1016/j.compag.2019.105162.

[16] J. G. A. Barbedo, "A new automatic method for disease symptom segmentation in digital photographs of plant leaves," *European Journal of Plant Pathology*, vol. 147, no. 2, pp. 349–364, 2016, doi: 10.1007/s10658-016-1007-6.

[17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, pp. 281-305, 2012, doi: 10.5555/2188385.2188395.

[18] S. Verma, A. Chug, A. P. Singh, S. Sharma, and P. Rajvanshi, "Deep Learning-Based Mobile Application for Plant Disease Diagnosis," *Advances in Environmental Engineering and Green Technologies Applications of Image Processing and Soft Computing Systems in Agriculture*, pp. 242–271, 2019, doi: 10.4018/978-1-5225-8027-0.ch010.

[19] F. J. P. Montalbo and A. A. Hernandez, "Classification of Fish Species with Augmented Data using Deep Convolutional Neural Network," *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, Shah Alam, Malaysia, 2019, pp. 396-401, doi: 10.1109/ICSEngT.2019.8906433.

[20] G. Castaneda, P. Morris, and T. M. Khoshgoftaar, "Evaluation of maxout activations in deep learning across several big data domains," *Journal of Big Data*, vol. 6, no. 1, Mar. 2019, doi: 10.1186/s40537-019-0233-0.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014, available at: https://arxiv.org/abs/1409.1556.

[22] C. Szegedy et al., "Going deeper with convolutions," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

[23] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195

[24] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778, doi: https://doi.org/10.1109/CVPR.2016.90.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Europ. Conf. Comput. Vision*, 2016, pp. 630–645, doi: 10.1007/978-3-319-46493-0_38.

[26] R. Atienza, "Advanced deep learning with Keras: apply deep learning techniques, autoencoders, GANs, variational autoencoders, deep reinforcement learning, policy tradients, and more", *Birmingham: Packt*, p. 59, 2018, available at: https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788629416/2/ch02lvl1sec13/resnet-v2.

[27] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in Proc. 8th Int. Conf. Quality Multimedia Exper. (QoMEX), Jun. 2016, pp. 1–6, doi: 10.1109/QoMEX.2016.7498955.

[28] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," Int. J. Remote Sens., vol. 28, no. 5, pp. 823–870, 2007, doi: 10.1080/01431160600746456.

[29] K. K. Pal and K. S. Sudeep, "Preprocessing for image classification by convolutional neural networks," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 1778-1781, doi: 10.1109/RTEICT.2016.7808140.

[30] B. L. Boese, P. J. Clinton, D. Dennis, R. C. Golden, and B. Kim, "Digital image analysis of Zostera marina leaf injury," *Aquatic Botany*, vol. 88, no. 1, pp. 87–90, 2008, doi: 10.1016/j.aquabot.2007.08.016.

[31] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in Proc. Eur. Conf. Comput. Vis., 2016, pp. 75–91, doi: 10.1007/978-3-319-46448-0_5.

[32] X. Zhang, Y. Qiao, F. Meng, C. Fan and M. Zhang, "Identification of Maize Leaf Diseases Using Improved Deep Convolutional Neural Networks," in *IEEE Access*, vol. 6, pp. 30370-30377, 2018, doi: 10.1109/ACCESS.2018.2844405.

[33] A. Krizhevsky, A., and G. Hinton, "Convolutional deep belief networks on cifar-10," *Unpublished manuscript*, vol. 40, no. 7, pp. 1-9, 2010, available at: https://www.cs.toronto.edu/-kriz/conv-cifar10-aug2010.pdf.

[34] R. Chauhan, K. K. Ghanshala and R. C. Joshi, "Convolutional Neural Network (CNN) for Image Detection and Recognition," *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, 2018, pp. 278-282, doi: 10.1109/ICSCCC.2018.8703316.

[35] F. Chollet, Deep Learning With Python. Shelter Island, NY, USA: Manning, 2018, pp.138-140, available at: Google Scholar.

[36] K. Kobayashi, J. Tsuji and M. Noto, "Evaluation of Data Augmentation for Image-Based Plant-Disease Detection," *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan*, 2018, pp. 2206-2211, doi: 10.1109/SMC.2018.00379.

[37] N. McLaughlin, J. M. Del Rincon and P. Miller, "Data-augmentation for reducing dataset bias in person re-identification," *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Karlsruhe, 2015, pp. 1-6, doi: 10.1109/AVSS.2015.7301739.

[38] G. C. Cawley and N. L. Talbot, "Over-fitting in model selection and subsequent selection bias in performance evaluation," *J. Machine Learn. Res.*, vol. 11, pp. 2079–2107, Jul. 2010, doi: 10.5555/1756006.1859921.

[39] T. Badriyah, D. B. Santoso, I. Syarif, and D. R. Syarif, "Improving stroke diagnosis accuracy using hyperparameter optimized deep learning," *International Journal of Advances in Intelligent Informatics*, vol. 5, no. 3, p. 256, 2019, 10.26555/ijain.v5i3.427.

[40] S. Nazir, S. Patel and D. Patel, "Hyper Parameters Selection for Image Classification in Convolutional Neural Networks," *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Berkeley, CA, 2018, pp. 401-407, doi: 10.1109/ICCI-CC.2018.8482081.

[41] N. S. Keskar and R. Socher. (Dec. 2017). "Improving generalization performance by switching from Adam to SGD," [Online]. Available: https://arxiv.org/abs/1712.07628

[42] H. Tercan, A. Guajardo, J. Heinisch, T. Thiele, C. Hopmann, and T. Meisen, "Transfer-learning: Bridging the gap between real and simulation data for machine learning in injection molding," Procedia CIRP, vol. 72, pp. 185–190, 2018, doi: 10.1016/j.procir.2018.03.087.

[43] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345-1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[45] L. Shao, F. Zhu and X. Li, "Transfer Learning for Visual Categorization: A Survey," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 1019-1034, May 2015, doi: 10.1109/TNNLS.2014.2330900.

[46] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning". Cambridge, MA, USA: *MIT Press*, 2016, available at: https://www.deeplearningbook.org/.

[47] R. Y. Rubinstein and D. P. Kroese, "The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning", *Springer Science & Business Media*, 2013, doi: 10.1007/978-1-4757-4321-0.

[48] D. M. Hawkins, "The Problem of Overfitting," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 1, pp. 1–12, 2004, doi: 10.1021/ci0342472.

[49] J. G. A. Barbedo, "Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification," *Computers and Electronics in Agriculture*, vol. 153, pp. 46–53, 2018, doi: 10.1016/j.compag.2018.08.013.

[50] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. Data Mining Knowl. Manage. Process*, vol. 5, no. 2, pp. 1–11, 2015, doi: 10.5121/ijdkp.2015.5201.

[51] Q. Liang, S. Xiang, Y. Hu, G. Coppola, D. Zhang, and W. Sun, "PD 2 SE-Net: Computer-assisted plant disease diagnosis and severity estimation network," Comput. Electron. Agric., vol. 157, pp. 518–529, Feb. 2019, doi: 10.1016/j.compag.2019.01.034.