Convolutional neural networks for the recognition of diseases and pests in Cassava leaves (*Manihot esculenta*)

Santiago María Gómez-Pupo[1]
Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar
Cartagena de Indias-Bolívar, Colombia.

Mary Andrea Fennix Agudelo[4]
Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar
Cartagena de Indias-Bolívar, Colombia.

Alberto Patiño-Vanegas[6]
Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar
Cartagena de Indias-Bolívar, Colombia.
apatino@utb.edu.co

Alberto Patiño-Saucedo[2, 3]
[2]*Electrical Engineering Department, Universidad de Guanajuato*. Guanajuato, México
[3]*Institute of microelectronics* Sevilla, España

Edisson Chavarro Mesa[5]
Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar
Cartagena de Indias-Bolívar, Colombia.

Abstract— *In this work, we recognized cassava diseases and pests, by means of convolutional neural networks, as a way to avoid the spread of pathogens, prevent economic losses, and favor decision-making for a proper disease management. For the development of this system, VGG16, ResNet50 and Xception models were chosen for having displayed good performance in previous researches of disease classification in plants, which we considered very similar to our case of study. For the training procedure, a transfer learning technique was implemented, employing a database categorized by cassava diseases (bacterial blight, brown streak, green mite, mosaic disease), as well as healthy leaves. This database was balanced and refined manually, selecting the images that represented characteristics of each category, according to the description found in the existing literature. Finally, the best model was chosen taking into account its performance measured through the Accuracy metric. The best model obtained, which we propose in this work, was Xception, and was trained during a period of 35 epochs with 6120 images of cassava leaves, achieving an accuracy of 94.56%. This model provides an option to detect cassava leaf diseases early, reliably and cost-effectively.*

Keywords— Deep learning, artificial vision system, plant disease.

I. Introduction

According to the Food and Agriculture Organization of the United Nations (FAO), cassava is the fourth most important basic product in the diet of more than 1 billion people, becoming one of the most significant sources of energy in tropical regions of the world [1]. However, this food security is affected by diseases and/or pests, causing losses in crop yield [2].

The phytopathogens frequently described and known for this crop worldwide are Cassava Brown Stripe disease (CBSD), Cassava Mosaic disease (CMD), Cassava bacterial blight (*Xanthomonas axonopodis* pv. manihotis) (CBB), and the Cassava green mite (*Mononychellus tanajoa*, m. caribbeanae) (CGM) [3].

Meanwhile, in Colombia, there are reports of the presence of these phytopathogens in the national territory [2], [4]. One of the strategies to successfully combat these diseases and pests is to guarantee early detection, which would allow farmers to establish the necessary actions for their proper management and control [5]. Usually, traditional methodologies of early diagnosis require personnel with observation skills to spot the diseases characteristics in crops. Nonetheless, in some cases the symptoms may be insufficient to establish an adequate diagnosis, thus laboratory analysis are required [6]. In general, the complexity of some recognition methods, and the availability of people able to quickly identify cassava diseases, make their monitoring and control challenging. Currently, due to technological advances, there are alternative methods such as artificial vision systems, which, together with Deep Learning (DL), are used successfully for automatic recognition of plant diseases. Diagnoses provided by these types of tools are reliable and allow farmers to make decisions about the appropriate treatment to implement. [7]. DL techniques, in particular convolutional neural networks (CNN), are now becoming an adequate method to detect pests and diseases, taking into account their great capacity to identify objects and establish patterns in images, thus optimizing crop management [8]–[11].

Diseases and/or pests identification from cassava crops images presents several challenges: (i) diseases characteristics can vary greatly depending on the disease state, stage, or progression degree (ii) more than one disease may occur in the same crop, or the diseases may have very similar characteristics, leading to misperceptions, which could represent a difficulty for the system; (iii) in some cases, it can be difficult to recognize a disease in its early stage and may be mistaken for a healthy leaf by the system.

Recent research has used CNN to detect cassava diseases and/or pests from images. These works have been carried out

using the database available in Kaggle competitions with 5 categories (CBB, CBSD, CGM, CMD, and healthy), including healthy leaves: Sambasivam G. et al [12], developed a CNN built from scratch and achieved 93% accuracy; Ayu, H. R. et al [13] used the MobileNetV2 architecture and obtained an accuracy of 65.6%, being the category of healthy sheets the most accurate of the model. In our research, using the databases available in https://www.kaggle.com/, and the same categories, a computer vision system was proposed to detect and identify cassava diseases, early and reliably, using CNN. The three networks VGG16, ResNet50 and Xception with transfer learning were implemented, being Xception the one that stood out for achieving the best performance, obtaining an accuracy of 94.56%, proving to be a good model for disease recognition in cassava leaves. However, in this work, unlike previous studies, we did not only limit to reporting the performance of the evaluated models, but a behavior analysis of the best model was carried out, based on each of the challenges described: (i) Which one of the diseases is more difficult for the model to identify, (ii) how well the model manages to recognize diseases that have similar characteristics, (iii) how much it accomplishes to spot diseases in their initial state, and (iv) how it performs when there is more than one disease in an image.

II. METHODOLOGY

In this work we used Convolutional Neural Networks (CNN) to determine if a cassava plant was healthy or had some disease, from an image of the plant, initially obtained in a public database (Fig. 1). CNNs were used for the complex problem of classifying diseases in plants, since they extract important characteristics without human supervision [14], which reduces the time consuming task of extracting features that generates erroneous results, thus, making it more accurate and efficient. Also, their training with transfer learning can improve computational performance by speeding up training time reusing models prepared in similar tasks. Furthermore, as in our case, transfer allows the use of less data in the preparation of a neural network compared to training from scratch, which requires large amounts of data [8].
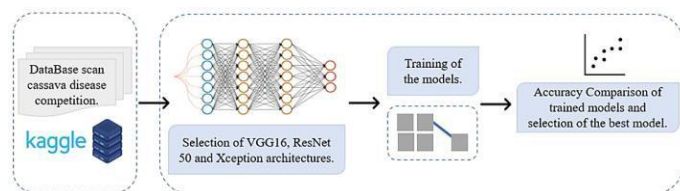

Fig. 1. Convolution Neural Network, Proposed methods for image classification of cassava diseases.

For training, we used data from two competitions held on the website https://www.kaggle.com/: one on April 26, 2019 that was part of the "Challenge of detailed visual categorization of cassava" (iCassava 2019 Fine-Grained Visual Categorization Challenge) and another on November 19, 2020 called "Classification of the yucca leaf disease" (Cassava Leaf Disease Classification). Both datasets contained images of cassava leaf diseases and images of healthy leaves, for a total of five categories: Cassava Brown Stripe disease (CBSD), Cassava Mosaic disease (CMD), Cassava bacterial blight (*Xanthomonas axonopodis* pv. manihotis) (CBB), Cassava green mite (*Mononychellus tanajoa*, m. caribbeanae) (CGM) y healthy leaves (Healthy). These databases, in addition to being unbalanced, contained repeated images, and also some images that do not corresponded to cassava leaves (stems or roots). The problem with unbalanced datasets is that classification learning algorithms often skew their predictions towards the majority class, thus, there is a higher rate of misclassification for the minority class [15]. Thus, we first performed a manual debugging, and then we took images of both databases to balance the classes. Finally, each of the five categories was made up of 1349 images, for a total of 6745 images. Of these, 80% were randomly selected for training, 10% for validation and 10% for testing [16].

For the development of the recognition system, three CNN architectures were selected: VGG16, ResNet50 and Xception, with a transfer learning approach. The selection of these architectures was based on their good performance in similar plants disease classification problems [8], [9], [11].

Trainings were carried out on a computer with 24 GB of Ram, 3GB GTX 1050 GDDR5 video card, and a 9th generation Intel Core i5 processor. The programming language was Python 3.8.5, and the library used was Tensorflow version 2.3.

The common hyperparameters and techniques for all models were: (i) a Bach size of 8 due to GPU memory limitations; (ii) a data augmentation technique with the following arguments: rotation range 45, zoom range 20%, vertical flip, horizontal flip, height shift range 10%, width shift range 10%, shear range 10% and fill mode nearest; (iii) a transfer learning technique using imagenet weights, where the last layer of each pre-trained model is exchanged for an average pooling layer to reduce dimensionality, a flatten layer that converts the output in a one-dimensional vector and a hidden dense layer with 1024 neurons, followed by a hidden dense layer with 512 neurons, both with a ReLu activation function and L2 regularization of 0.01. Finally, (v) an output layer with 5 neurons and a Softmax activation function is also included; an Adam optimizer and a Dispersed Categorical Cross Entropy loss function.

For the VGG16 and ResNet50 models, the transfer learning was implemented in two phases. In the first phase, the previously trained weights were loaded and their layers were frozen to train only the output layer and a learning rate of 1.0×10^{-3} is used. In the second phase, a fine adjustment of the models is carried out, for VGG16 the layers of block 5 are trained. For ResNet50, the normalization layers were updated in batches and the layers are trained from 165 onwards, both with a learning rate of 1.0×10^{-5}.

For the Xception model, the transfer of learning technique is implemented in a single phase and the best results were obtained when performing Fine-Tuning, training the blocks layers 13 and 14 with a learning rate of 1.0×10^{-3}.

The metrics that we implemented in this work to evaluate CNN performance are described below. Accuracy: This metric measures the percentage of correctly classified data from both positive and negative classes. It is important to keep in mind that accuracy does not work well when classes are unbalanced. Precision: This metric measures the percentage of the data that are from the positive class, with respect of all the examples that the model has classified as positive. Recall: For each class, this metric measures the percentage of positive cases that were correctly identified by the model.

The balanced database used for model training can be found at: https://www.kaggle.com/disonant/cassava-diseases-balanced/metadata. The codes necessary to replicate the training of the models implemented in this work are available in the following GitHub repository: https://github.com/Sangop21/Deep-learning-Cassava-Diseases.

III. RESULTS

We present below the results of the training, validation and testing of the three selected models VGG16, ResNet50 and Xception. For their training, the same database described in the previous section was used, and all were evaluated with the same test set (625 images not used either for training or for validation). To choose the model with the best performance, the highest value of the Accuracy metric was used as a criteria, which measured the percentage of correct predictions, being reliable because the classes are balanced.

Table 1 compares the three models according to the results obtained when evaluating them with the test dataset. We observed that Xception model outperforms the other pre trained models with an accuracy of 94.55%, and a loss of 0.2058. Furthermore, it is observed in Fig. 2 that approximately from epoch 8, the Xception model has an Accuracy of more than 0.9 and a loss approximately less than 0.3.
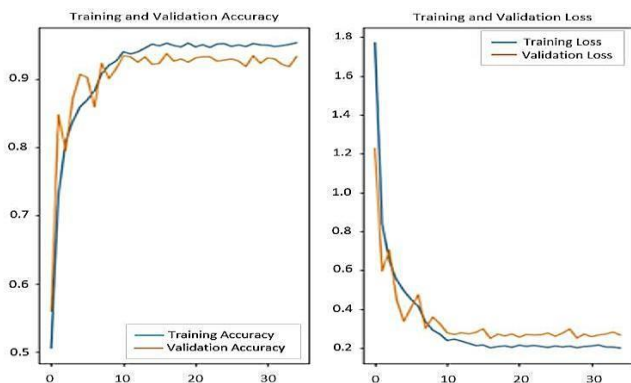

Fig. 2. Learning curve of the Xception model with Transfer-learning and Fine-tuning. Left: Accuracy, Right: Loss.

Table 1. Results of the evaluation of each model with the test set.

Model	Accuracy (%)	Loss
Model 1: VGG16	87.18	0.5046
Model 2: ResNet50	92.95	0.2927
Model 3: Xception	94.55	0.2058

Figure 3 shows the confusion matrix obtained in the classification of the 625 images that are part of the test set, and Table 2 shows the performance metrics calculated from it.
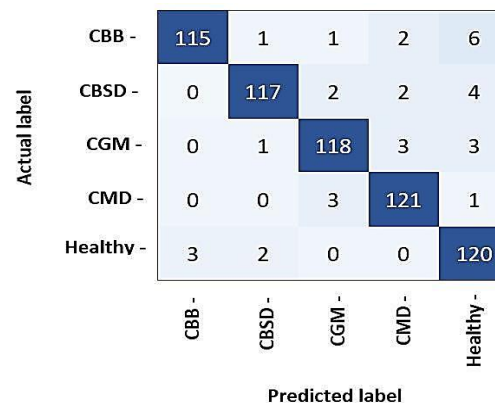

Fig. 3. Confusion matrix of the Xception from 625 images.

We carried out a more detailed analysis of the performance on the Xception model from the following aspects: (i) Which diseases the model finds most difficult to identify, (ii) how well the model manages to recognize the diseases that have characteristics similar or in their initial state, (iii) how the model behaves when there is more than one disease in an image.

Table 2. Xception model performance metrics.

Category	Precision	Recall	Support
Cassava bacterial blight (CBB)	0.9746	0.9200	125
Cassava Brown Stripe disease (CBSD)	0.9669	0.9360	125
Cassava green mite (CGM)	0.9516	0.9440	125
Cassava Mosaic disease (CMD)	0.9453	0.9680	125
Healthy	0.8955	0.9600	125
Accuracy		0.9456	625
Macro average	0.9468	0.9456	625

The metric used to determine which diseases were difficult for the system to recognize was Recall, since it gave us an idea of the capacity that the model had to detect a certain disease in the leaves that effectively present it. Table 2 shows that bacterial blight (CBB) and brown streaking (CBSD) were the most difficult categories for the model to classify, since they had less completeness. 8% of the images with bacterial blight, and 6.4% of the images with brown stripes were not identified

by the model, these values being the highest compared to the rest of the categories. (ii) To determine how much the model managed to recognize diseases that have similar characteristics, we analyzed the Precision metric, since it allowed us to determine the percentage of positive cases detected in each class; and furthermore, define how good the model was at avoiding false positives. In Table 2 we can see the value of the average macro of the precision, which indicates that the model identified the different categories with a performance of 94.68%, even though they had similar characteristics. As an example, Fig. 4 shows two diseases of the cassava leaf, the green mite and the mosaic, where we can see that in some of their advanced stages these diseases have a lot of similarity. However, the Precision value for the green mite category indicates that the model correctly avoids false positives in this category by 95.16%. Similarly, the model eludes false positives of the mosaic category by 94.53%.
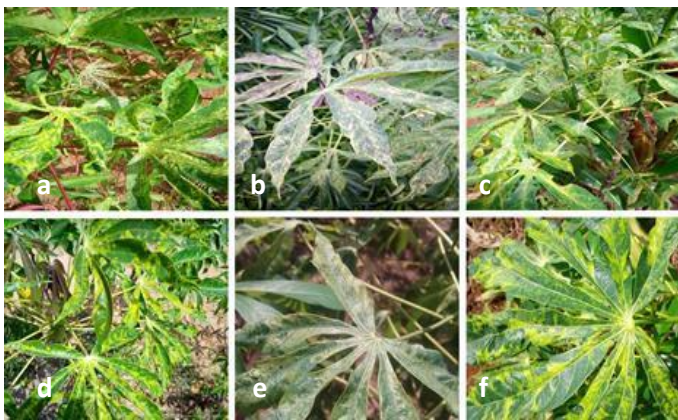

Fig. 4. Similar characteristics between the green mite and the yucca mosaic; a, b, c (Cassava green mite symptoms - CGM); d, e, f (Cassava mosaic symptoms - CMD).

It was analyzed how much the model managed to identify diseases in the cassava leaf when first symptoms hardly appeared. Sometimes, in their initial state, diseases can go unnoticed, since the symptoms are not very noticeable as when the disease is in an advanced stage, hence being confused with healthy leaves. Thus, in order to determine the performance of the system to detect diseases in their initial stage, it was proposed to count the false positives in the Healthy leaves category, and from those false positives, the performance of the model was determined in hitting the second highest probability. According to the confusion matrix, of the 14 false positives in the Healthy category, 11 images in the model hit the category with the second highest probability. Therefore, we can conclude that the performance of the model in detecting diseases in their early stage was 78.6%.

In Figure 5 we have a sample of images that the model mistook for healthy leaves. We can observe, according to the probabilities that the model assigns, that the second option to perform the classification is the correct category. That is, although the model performed a classification as a healthy leaf, it is capable of detecting the actual disease.

On the other hand, Fig. 6 shows some images with diseases in their initial stage that the model correctly detected (first higher probability). This indicates that the system performs well in detecting diseases in their early stage, either in the first option or in the second option according to their probability of success.
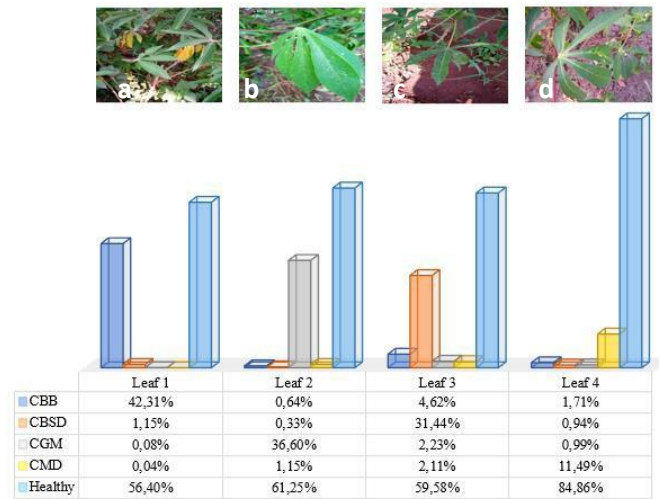

	Leaf 1	Leaf 2	Leaf 3	Leaf 4
CBB	42,31%	0,64%	4,62%	1,71%
CBSD	1,15%	0,33%	31,44%	0,94%
CGM	0,08%	36,60%	2,23%	0,99%
CMD	0,04%	1,15%	2,11%	11,49%
Healthy	56,40%	61,25%	59,58%	84,86%

Fig. 5. Some images mistaken for healthy leaves. (a) Image with real category of bacterial blight. (b) Image with brown striped real category. (c) Image with real green mite category. (d) Image with real mosaic category.

Fig. 6. Diseases in their initial state with their respective percentage of correct prediction; a) CBB (99,8%), b) CBB (91,7%), c) CBSD (99,1%), d) CBSD (96,6%), e) CGM (99,3%), f) CGM (97,7%), g) CMD (99,8%), h) (99,3%).

Next, we analyzed the behavior of the model when more than one disease was present in an image. Although there is no category in the database that indicated the presence of more than one disease, we took a small sample of images that were not used during model training. These were processed manually by superimposing on one of the images from one category, clippings from the image from another category. In this way, we obtained three images where with certainty we know that there are two diseases of the cassava leaf.

Figure 7 shows the three images obtained. In Fig. (a) brown striped (CBSD) and green mite (CGM) were mixed, in Fig. (b) green mite (CGM) and mosaic (CMD) and in Fig. (c) brown striped (CBSD) and brown striped (CBSD). In the images, the area where each of the diseases occurs was enclosed in different colors. We observe that the two highest probabilities are indeed the two categories that we mix in the processed images. Therefore, it is evident that the system could be capable of detecting the presence in an image of more than one disease.
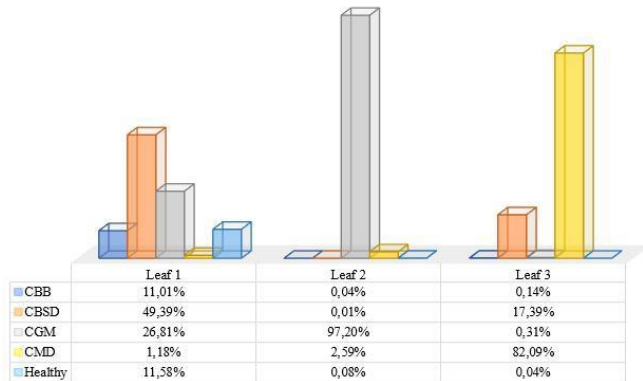
Fig. 7. Images with the presence of two diseases. (a) Brown striped and green mite. (b) Green and mosaic mite. (c) Brown striped and mosaic.

	Leaf 1	Leaf 2	Leaf 3
CBB	11,01%	0,04%	0,14%
CBSD	49,39%	0,01%	17,39%
CGM	26,81%	97,20%	0,31%
CMD	1,18%	2,59%	82,09%
Healthy	11,58%	0,08%	0,04%

IV. CONCLUSIONS

The Xception model trained in this work applying transfer learning and fine tuning, with an accuracy of 94.55%, is proposed to determine if a cassava plant is healthy or has any disease (bacterial blight, brown streaking, green and mosaic mite), from an image of the plant captured in the field. According to the study carried out, the model showed good performance in recognizing diseases in their initial state, discriminating diseases even if they have similar characteristics, and detecting the presence of more than one disease in the same image.

REFERENCES

[1] J. Aristizábal, T. Sánchez, and D. M. Lorío, *Guía técnica para producción y análisis de almidón de yuca*. Organización de las Naciones Unidas para la Agricultura y la Alimentación …, 2007.

[2] B. V. Herrera Campo, G. Hyman, and A. Bellotti, "Threats to cassava production: known and potential geographic distribution of four key biotic constraints," *Food Secur.*, vol. 3, no. 3, pp. 329–345, 2011.

[3] V. Alonso Chavez, A. E. Milne, F. van den Bosch, J. Pita, and C. F. McQuaid, "Modelling cassava production and pest management under biotic and abiotic constraints," *Plant Mol. Biol.*, 2021.

[4] Departamento Administrativo Nacional de Estadística - DANE, "El cultivo de la yuca (Manihot esculenta Crantz)," *Boletín Mens. insumos y factores Asoc. a la Prod. Agropecu.*, vol. 46, pp. 1–7, 2016.

[5] B. Ospina and H. Ceballos, "La yuca en el tercer milenio." 2002.

[6] M. B. Riley, M. R. Williamson, and O. Maloy, "Plant disease diagnosis," *Plant Heal. Instr.*, vol. 10, 2002.

[7] S. M. Hassan, A. K. Maji, M. Jasiński, Z. Leonowicz, and E. Jasińska, "Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach," *Electronics*, vol. 10, no. 12, p. 1388, 2021.

[8] L. Mkonyi *et al.*, "Early identification of Tuta absoluta in tomato plants using deep learning," *Sci. African*, vol. 10, p. e00590, 2020.

[9] T. Anandhakrishnan and J. S. M. Murugaiyan, "Identification of Tomato Leaf Disease Detection using Pretrained Deep Convolutional Neural Network Models," *Scalable Comput. Pract. Exp.*, vol. 21, no. 4, pp. 625–635, 2020.

[10] Y. Toda and F. Okura, "How convolutional neural networks diagnose plant disease," *Plant Phenomics*, vol. 2019, 2019.

[11] J.-R. Xiao, P.-C. Chung, H.-Y. Wu, Q.-H. Phan, J.-L. A. Yeh, and M. T.-K. Hou, "Detection of Strawberry Diseases Using a Convolutional Neural Network," *Plants*, vol. 10, no. 1, p. 31, 2021.

[12] 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.

[13] H. R. Ayu, A. Surtono, and D. K. Apriyanto, "Deep learning for detection cassava leaf disease," in *Journal of Physics: Conference Series*, 2021, vol. 1751, no. 1, p. 12072.

[14] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, 2021.

[15] A. N. Tarekegn, M. Giacobini, and K. Michalak, "A review of methods for imbalanced multi-label classification," *Pattern Recognit.*, vol. 118, p. 107965, 2021.

[16] B. Genc and H. Tunc, "Optimal training and test sets design for machine learning," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 2, pp. 1534–1545, 2019.