Development of a Convolutional Neural Network Architecture for Classifying Foliar Diseases in Plants

Roney Nogueira de Sousa[1] **Ronieri Nogueira de Sousa** [2]

[1] Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Av. Tabelião Luiz Nogueira de Lima S/N – Tianguá – CE – Brazil

[2]Universidade Estácio de Sá
Rua 31 de Julho Nº 571 – Tianguá – CE – Brazil

{nogueiraroney453,nsronieri}@gmail.com

Abstract. *This study aims to build and test a CNN architecture on a database extracted from the PlantVillage platform. The database contains 54,305 images, divided into 39 classes, with 38 classes related to foliar diseases and one class for background images. Due to the small amount of images in each class, Data Augmentation techniques were used to generate additional synthetic images during the network training process. The results achieved were an accuracy of 99.55%, precision of 94.25%, AUC of 99.81% and F1-Score of 90.86%.*

Resumo. *Este estudo tem como objetivo construir e testar uma arquitetura de RNC em um banco de dados extraído da plataforma PlantVillage. O banco de dados contém 54.305 imagens, divididas em 39 classes, sendo 38 classes relacionadas a doenças foliares e uma classe para imagens de fundo. Devido à pequena quantidade de imagens em cada classe, técnicas de Data Augmentation foram utilizadas para gerar imagens sintéticas adicionais durante o processo de treinamento da rede. Os resultados alcançados foram uma acurácia de 99,55%, precisão de 94,25%, AUC de 99,81% e F1-Score de 90,86%.*

1. Introduction

Convolutional Neural Networks (CNN) are networks specialized in data processing with a grid-like topology. For example, image data has its representation in a 2D grid of pixels or time series data that can be taken as 1D grid samples of a regular period. In addition, CNN can be applied to different types of data with different dimensions [Goodfellow et al. 2016]. Deep learning provided a future with openness to several increments in several domains, including the field of agriculture, among which there is a possibility of growth.

CNNs are used in the processing of images, videos, voice, and audio. CNNs have been increasingly used in image classification, for example, for medical images, road signs, objects and other fields of use. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has helped the constant evolution of CNNs regarding the evaluation of different architectures using an extensive database with different categories of objects [Marques 2016, Souza et al. 2020].

One of the most important tasks in agricultural practices is the detection of diseases in crops, as it is essential to avoid losses in production. However, this task presents the problem of requiring a lot of time and a qualified workforce.

In order to assist professional of the area in the diagnosis of various diseases, Diagnostic Aid Systems are developed. Usually, these systems use image processing and machine learning techniques to classify images as healthy or diseased, classifying them in one of the disease classes thus providing a supplementary diagnosis. They can also assist in serving large audiences, thus playing the role of a tool to prevent fatigue on the part of the specialist, which can lead to erroneous diagnoses.

The work of Marques Junior [Marques Junior 2019] can be highlighted, in which the authors sought to carry out the classification of five species of weeds found throughout the national territory, whereas in the work of Silva and Schimiguel [Silva and Schimiguel 2020], the authors aimed to identify plant diseases in the EMBRAPA database.

As previously seen, a study will be carried out, aiming at the proposal of an CNN architecture to carry out the diagnosis of the plant. The aim is to analyze the results based on quantitative procedures, with emphasis on network training and testing. For training and testing the network, the public database available on the PlantVillage[1] will be used. This dataset consists of 54,305 images.

However, one of the problems that can arise when using these networks is overfitting, making it difficult to generalize features to classify new images. Although it is not an exclusive characteristic of CNNs, it can occur in these networks due to their great modeling capacity. Because CNNs have many trainable parameters, they can overfit the training data and perform worse on unseen data [Goodfellow et al. 2016].

The data augmentation technique was used, performing image transformations, with the purpose of mitigating the overfitting of the model, since during the network training process, some images undergo transformations, in order to be able to expose the model to different images [Li et al. 2021].

This article proposes a CNN model, which provides an accuracy of 99.55% using 39 different classes of plant leaves. The model is able to automatically classify the type of disease the plant is affected by, bringing with it the possibility of saving time and increasing diagnostic efficiency.

1.1. Document Organization

This study is organized in 6 Sections. Section 2 provides a literature review of works similar to the present study, providing a greater scientific base. Section 3 provides the idea of the proposed architecture of the present study. Section 4 presents the methodology adopted in this work, addressing the adopted database, the development environment and data preparation. Section 5 shows the results achieved by the proposed architecture and their due discussions. Section 6 discusses the conclusions obtained from the results of the experiments and presents suggestions for future work.

2. Related Works

This literature review aims to demonstrate the effectiveness of commonly used machine learning approaches in addressing various goals. Machine learning offers a flexible and

[1] https://plantvillage.psu.edu/

powerful framework for decision-making and integrating expert knowledge. These advantages make machine learning algorithms widely utilized in various fields, including agricultural mechanization.

The study by Funck [Funck 2019] introduced a mobile application on the Android platform. This app utilized the smartphone camera to identify Asian soybean rust on leaves. A total of 2,770 soybean leaf images were collected, with a portion used for testing and another for validation. The application achieved an accuracy of 84.61% in detecting Asian rust.

The system developed by Afonso [Afonso 2019] utilizes computer vision techniques to diagnose diseases in grapevine leaves. The system focuses on extracting specific characteristics of a particular disease for future identification. By employing specific capture conditions, they achieved a leaf state diagnosis accuracy of 95.17%.

In their study, Hasan et al. [Hasan et al. 2020] evaluated the performance of nine deep learning models for plant disease classification. They employed two approaches: transfer learning, where the last three layers of the networks were replaced, and feature extraction from specific layers of the models used as input for machine learning classifiers. The study reported a maximum accuracy of 97.45

In leaf disease classification, several machine and deep learning techniques have been proposed. However, a common challenge is the large number of model parameters. This study addresses this issue by proposing a CNN model with fewer parameters than existing approaches. To prevent overfitting, data augmentation techniques and network dropout were used. These strategies improve the model's generalization and performance.

3. Convolutional Neural Network Architecture

The proposed network architecture is composed of ten Convolutional layers, eight Batch Normalization layers, six Max Pooling layers, five Dropout layers, four Dense layers and one Flatten layer. Figure 1 shows the architecture proposed in this study.

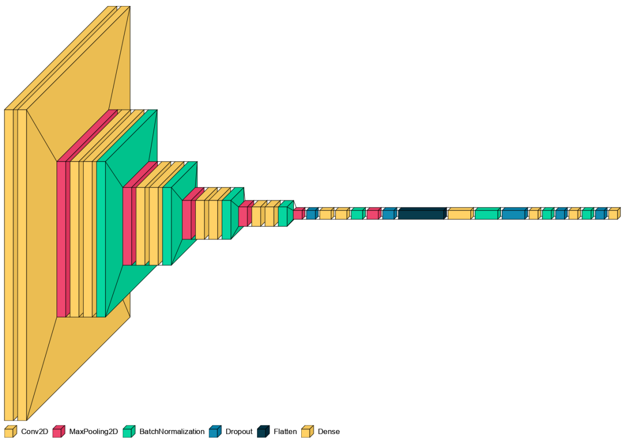

Figure 1. Proposed CNN Architecture

At the end of the architecture, the dense layers were added, they were alternated with Batch Normalization layers and with the last three levels of dropout. At the end of the network, a dense layer with the softmax activation function was added.

3.1. Layers Present in the Proposed Architecture

The first layer in the model is the input layer, in this layer the images are normalized and then fed into the model. The convolutional layers are responsible for receiving an input image, grouping weight filters and adding bias values to generate a feature or response map. The resource's response is then forwarded to the next subsequent layer. Max Pooling layers are used on each entry to lower the computational cost of subsequent layers by choosing the maximum pixel value in the chosen kernel. Max Pooling is calculated using Equation 1.

$$MaxPooling = \frac{I_x - P}{S} + 1 \tag{1}$$

It can be seen as follows:

- I_x: Input;
- P: Pooling window size;
- S: Step

Dropout layers are used to randomize the proposed method to randomly drop some neurons in the hidden layer during training. Dropout is also the regularization technique used to reduce the overfitting problem in the model. After the convolutional layers, the flatten layer is used to convert the high-dimensional data into a single column vector.

The dense layers connect each neuron from the previous layer to neurons in the dense layer. Following the block of dense layers, a final dense layer with the SoftMax activation function is utilized. This last layer has a number of neurons equal to the number of classes for classification purposes.

3.2. Function Loss

The Categorical Cross-Entropy [Zhang and Sabuncu 2018] function was used as a loss function to calculate the probability of each image in the classes. The equation of this function is demonstrated in Equation 2.

$$CategoricalCross - Entropy = \frac{\partial}{\partial_{S_P}} \left(-\log \left(\frac{e^{s_p}}{\sum_j^c e^{S_j}} \right) \right) \tag{2}$$

In the multiclass classification, the labels used are of the one-hot encoding type and there are only positive classes in the loss term. S_p is the positive class score. The gradient calculations performed by the CNN output neurons result in a backpropagation through the network that optimizes the loss function.

3.3. Optimization Algorithm

The Root Means Square Propagation [Riedmiller and Braun 1993] optimizer was also used as the optimization algorithm to train the proposed model. It has an adaptive learning rate instead of treating the learning rate as a hyperparameter. The learning rate update rule is exposed by Equations 3, 4 and 5.

$$v_t = \gamma v_{t-1} + (1 - \gamma) * g_t^2 \tag{3}$$

$$\Delta w_t = -\frac{\eta}{\sqrt{v_t + \epsilon}} * g_t \tag{4}$$

$$w_{t+1} = w_t + \Delta w_t \tag{5}$$

4. Methodology

This experiment aimed to train and test a new CNN architecture for leaf disease image classification. The focus was on evaluating the network's ability to identify and classify diseases in the images, and analyzing its evaluation metrics.

4.1. Development Environment

To implement the networks, the programming language Python [Van Rossum and Drake 2009] was used. For the development of the networks, the API Keras [Chollet et al. 2015] was used.

The Matplotlib [Hunter 2007] library was used to visualize the graphic images. Finally, the Visualkeras [Gavrikov 2020] library was used to visualize the produced CNN.

For this work, the configuration of the machine used in the experiments has the following technical specifications:

- AMD Ryzen 5 5600X, 4.6GHz with 6 physical cores and 12 threads;
- 32 GB RAM memory;
- 1.5 TB storage;
- NVIDIA GeForce RTX 2060 Super video card, with 8 GB of dedicated memory.

4.2. Database

The PlantVillage dataset is mainly composed of color images containing 39 different disease classes for the classification task. The dataset contains 55,448 images of 13 different plant species, which were used to train and test the CNN model proposed.

The database has 38 different classes, each class being defined as a healthy or infected plant, identifying the disease which infected the plant using labels. As mentioned earlier, one more class is also present and it has 1143 background images.

4.3. Preparation of Data and Classifiers

With the need to smooth overfitting effects, data pre-processing steps were carried out in the database used. As the amount of images for each class contained in the database is considerably small, the proposed methodology aims to apply data augmentation techniques. For the data augmentation process, the Albumentations [Buslaev et al. 2020] library was used.

For image transformations, inversion, gamma correction, noise injection, PCA color enhancement, rotation, and scaling techniques were used. The transformations were applied sequentially, so that each image could receive a different number of transformations. This way of increasing data was chosen because it allows controlling the number of variations generated and avoiding the generation of data that are very similar to each other.

Data augmentation was applied exclusively to the training set, as mentioned by Shorten and Khoshgoftaar [Shorten and Khoshgoftaar 2019]. This approach prevents the introduction of duplicate data into the test set, which could lead to an overestimation of the model's accuracy.

The proposed architecture in this work was trained for 1000 epochs, where each epoch involved applying all the training set images to the model. After each epoch, the model's current state was validated using the validation dataset, and the results were saved. Finally, the trained model's performance was assessed using the test set.

For the process of training and test of the architecture, the following callbacks functions were used:

- ***Early Stopping*** [2]**:** This callback is used to prevent overfitting in the model. In this study, the F1-Score metric was monitored, and the training was stopped if there was no improvement in this metric for 25 epochs;
- ***LearningRateScheduler*** [3]**:** With this callback, at the start of each epoch, the schedule function obtains the updated value of the learning rate. This function is predefined before training, using the current epoch and learning rate.

The Root Mean Square Propagation optimizer, described in Section 3.3, was used in conjunction with the Categorical Cross Entropy [4] loss function. During training, an initial learning rate of 1×10^{-3} was used, and this rate was adjusted to smaller values using the LearningRateScheduler callback.

5. Results and Discussions

After carrying out the network training process, it can be seen in Figure 2 that it had an accuracy of 99.50% during the training process and an AUC of 99.76%. It can also be seen in the same image how the process of evolution of the metrics was over the epochs. The architecture obtained satisfactory results and a low number of epochs for it to converge, taking only 25 epochs for the results of the F1-Score metric to stop improving.

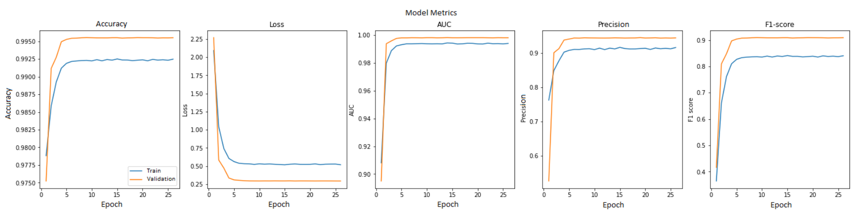

Figure 2. Results Obtained by the Proposed Network Architecture

In addition, it can also be seen in Figure 2 that during the model validation process, after 25 epochs, the loss function reached the lowest value (0.2621), during this epoch it was seen that the network managed to reach an accuracy corresponding value of 99.50%, precision of 94.13%, AUC with a value of 99.76% and, finally, an F1-Score of 90.86%.

[2]`https://keras.io/api/callbacks/early_stopping/`

[3]`https://keras.io/api/callbacks/learning_rate_scheduler/`

[4]`https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalCrossentropy`

The proposed CNN model performed satisfactorily during the classification task, the AUC and F1-Score indices indicate that the network was able to perform the classification task satisfactorily and with good generalization. The indices shown in Table 1 show how the behavior of the network results using the test set was given.

Table 1. Result Behavior

Metric	Value
Accuracy	99.55%
Precision	94.25%
AUC	99.81%
F1-Score	90.86%

The results achieved by the proposed network can be interpreted as follows. The model accuracy, reaching 99.55% during the model testing process, indicates that the architecture managed to classify most of the samples correctly. This implies that she can learn the distinct characteristics for each of the 38 classes of foliar diseases and identify what can be classified as background from the background image.

In addition to accuracy, precision obtained results with high rates, presenting a value of 94.25% during the network testing process. The value reached indicates that most of the samples classified as positive by the network were really positive, this result is important for the disease classification task, to avoid erroneous classifications that can be harmful to the health of the plants.

The AUC metric, which reached an index of 99.81%, indicates that the model was able to distinguish with high precision between positive and negative classes. Thus, it can be indicated that the proposed architecture was able to distinguish with high precision between the different classes of foliar diseases.

Finally, the F1-Score, which is a metric that combines precision and recall, obtained good results, 90.86% in the test set. This suggests that the network achieved a good balance between accuracy and recall, which is important to ensure that the network can correctly detect all positive samples, while minimizing the number of false positives.

6. Conclusion and Future Work

The present study proposed a CNN architecture for classifying leaf diseases using leaf images and different background images for network training. The dataset was obtained from the PlantVillage platform, and due to the limited number of images in each class, data augmentation techniques were applied to generate synthetic samples.

The developed model achieved outstanding performance in classifying leaf disease images with high accuracy. With an accuracy of 99.55%, precision of 94.25%, AUC of 99.81%, and F1-Score of 90.86%, the results demonstrate the success of the study in the classification task. These findings suggest that the model can serve as an effective tool to support specialists in the diagnostic process or act as a second opinion simulator.

However, it is important to mention that the study has limitations, such as the use of only one dataset for training and testing the classifier. For future works, it is suggested the use of other datasets, as well as the improvement of the model so that it can perform the classification task using less computational resources.

References

Afonso, S. (2019). Utilização de algoritmos de visão computacional para diagnóstico de doenças da vinha. Master's thesis, Universidade do Porto.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., and Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. *Information*, 11(2):125.

Chollet, F. et al. (2015). Keras. `https://keras.io`.

Funck, F. C. (2019). Detectando a ferrugem asiática na folha da soja utilizando redes neurais convolucionais. B.S. thesis, Universidade Tecnológica Federal do Paraná.

Gavrikov, P. (2020). visualkeras. `https://github.com/paulgavrikov/visualkeras`.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

Hasan, R. I., Yusuf, S. M., and Alzubaidi, L. (2020). Review of the state of the art of deep learning for plant diseases: A broad analysis and discussion. *Plants*, 9(10):1302.

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*.

Marques, E. A. L. (2016). Estudo sobre redes neurais de aprendizado profundo com aplicações em classificação de imagens. *Biblioteca Digital da Produção Discente UNB*.

Marques Junior, L. C. (2019). Classificação de plantas daninhas em banco de imagens utilizando redes neurais convolucionais. *Universidade Estadual Paulista (Unesp)*.

Riedmiller, M. and Braun, H. (1993). A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE international conference on neural networks*, pages 586–591. IEEE.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. In *Journal of Big Data*, volume 6, page 60. Springer.

Silva, M. J. and Schimiguel, J. (2020). Identificação de doenças em plantas por meio de processamento de imagens: Redes neurais convolucionais como auxílio à agricultura. *Revista de Ubiquidade*, 3(1):91–111.

Souza, V., Silva, L., Santos, A., and Araújo, L. (2020). Análise comparativa de redes neurais convolucionais no reconhecimento de cenas. *Anais do Computer on the Beach*, 11(1):419–426.

Van Rossum, G. and Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA.

Zhang, Z. and Sabuncu, M. (2018). Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31.