Comparing the Performance of Two Convolutional Neural Networks for Tomato Leaf Disease Classification

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. *Classifying diseases on tomato leaves, in particular, is a challenge for growers because it is difficult to visually identify the subtle differences between the symptoms of different diseases. Seeking to solve this problem, this study presents a comparative analysis of the performance of neural networks Inception and DenseNet in the classification process of diseases in tomato leaves using Transfer Learning and Data Augmentation techniques. Through experimental analysis, the DenseNet network has the best classification behavior, managing to classify diseases with 98.25% accuracy, 95.33% accuracy, 94.09% recall and AUC of 98.72%.*

Resumo. *A classificação de doenças em folhas de tomate, em particular, é um desafio para os agricultores porque é difícil identificar visualmente as diferenças sutis entre os sintomas de diferentes doenças. Buscando resolver este problema, este estudo apresenta uma análise comparativa do desempenho das redes neurais Inception e DenseNet no processo de classificação de doenças em folhas de tomate utilizando técnicas de Transfer Learning e Data Augmentation. Através da análise experimental, a rede DenseNet possui o melhor comportamento de classificação, conseguindo classificar doenças com 98,25% de precisão, 95,33% de precisão, 94,09% de recordação e AUC de 98,72%.*

1. Introduction

Tomato cultivation is of great economic importance worldwide, but plant diseases can lead to significant losses in production and quality. Early detection of diseases in plants can be a key factor in minimizing these losses, and machine learning techniques have been used for this purpose. In this study, a comparative analysis of the performance of Inception [Szegedy et al. 2016] and DenseNet [Huang et al. 2017] neural networks in the process of classifying diseases in tomato leaves using Transfer Learning and Data Augmentation techniques is presented.

The classification of diseases in plants is an important task for agriculture, as it can help prevent and control disease outbreaks, minimizing crop loss and increasing productivity. The classification of diseases in tomato leaves, in particular, is a challenge for farmers, as it is difficult to visually identify the subtle differences between symptoms of different diseases. The use of machine learning techniques, such as convolutional neural networks, can be a solution to this problem. Previous studies have shown that these techniques have been effective in classifying diseases in plants with high accuracy and efficiency [Fuentes and Yoon 2018, Mohanty et al. 2016, Sladojevic et al. 2016]. Furthermore, the use of Transfer Learning and Data Augmentation techniques has shown significant improvements in the generalization capability and accuracy of classification models [Pan and Yang 2010, Shorten and Khoshgoftaar 2019].

Images of tomato leaves infected with nine common diseases were used: Bacterial spot, Early blight, Late blight, Leaf Mold, Septoria leaf spot, Target Spot, Tomato mosaic virus, Tomato yellow leaf curl virus, Two-spotted spider mite, and a class of healthy leaves. These images were obtained from the online repository provided in [Huang and Chang 2020]. The images were divided into training and testing sets, and the Inception and DenseNet networks were trained and tested with these sets using different parameter configurations and regularization techniques.

The results were evaluated based on various performance metrics, including accuracy, AUC, precision, recall, and loss. These metrics provide important information about the quality of classification and the ability of neural networks to distinguish between different diseases and healthy leaves.

1.1. Document Organization

This article is organized into 6 Sections. In Section 2, the concepts and theoretical foundations necessary to understand the proposed work will be presented. In Section 3, academic works related to similar subjects as this present work will be reviewed. In Section 4, the methodology adopted for the development of the work will be presented. In Section 5, the results obtained with the proposed methodology will be presented, and an analysis of these results will be performed. In Section 6, the main conclusions reached from the analysis of the results will be discussed.

2. Theoretical Foundation

The theoretical foundation is a crucial aspect of any research study, as it provides the necessary understanding of the concepts and principles that underpin the proposed work.

2.1. Inception

Inception is a CNN that was first introduced by Google researchers in 2014 [Szegedy et al. 2015]. The Inception network is well known for its ability to efficiently train deeper and wider neural networks, which are crucial to achieving peak performance in various [Yosinski et al. 2014] computer vision tasks. Figure 1 shows the architecture of the Inception.

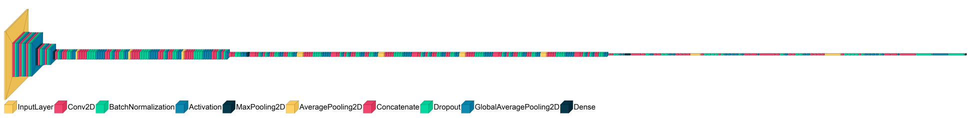

Figure 1. Convolutional Neural Network Architecture Inception

One of the main advantages of the Inception network is its ability to perform image classification tasks well while using less computational resources than other deep CNN [Xie et al. 2017] architectures. This is achieved through the use of various specialized modules, such as the initiation module, which allows the network to learn complex and diverse features with minimal computational cost.

2.2. DenseNet

DenseNet is a CNN architecture proposed by [Huang et al. 2017], which has achieved impressive results in various computer vision tasks, such as image classification and semantic segmentation. The main idea behind the DenseNet architecture is to connect all layers in a dense network, where each layer receives as input the outputs of all previous layers Figure 2 shows the architecture of the DenseNet.

Figure 2. Convolutional Neural Network Architecture DenseNet

One of the main advantages of DenseNet is its ability to deal with the problem of gradient fading in deep networks, which is one of the main challenges in training deep networks. In fact, dense connection between layers in a DenseNet network helps propagate the gradient throughout the network, leading to better stability during training.[Wang et al. 2018, Bhattacharjee et al. 2019]

3. Related Works

The literature review in this section aims to demonstrate how commonly used machine learning approaches efficiently handle this objectives.

Hong et al. [Hong et al. 2020] aimed to use transfer learning to make the process of building deep learning networks for detecting and diagnosing diseases in tomato leaves faster and more cost-effective. They used five different CNNs to extract features from nine types of tomato leaf diseases. Among the models tested, Densenet_Xception achieved the highest recognition accuracy at 97.10%, but it had more parameters. On the other hand, ShuffleNet achieved an accuracy of 83.68% with fewer parameters.

Zaki et al. [Zaki et al. 2020] work's aimed to identify diseases in tomato plants through leaf images. A compact MobileNet V2 was used to identify three types of diseases in tomato plants. The algorithm was tested on 4,671 images from the PlantVillage dataset and achieved a disease detection accuracy of over 95.94%.

A model based on a convolutional neural network was proposed in the work of Agarwal et al. [Agarwal et al. 2020]. The model is compared with other pre-trained models and the experimental results show that the proposed model is effective in detecting and classifying nine tomato diseases, with an average accuracy of 91.2%.

Based on what has been exposed, it can be stated that regardless of the nature of the research carried out, the researchers resorted to digital tools and aimed to ensure that the chosen data were later analyzed carefully and systematically. This makes the results obtained more reliable. In addition, it is worth mentioning that the related works were analyzed with the greatest affinity in relation to the problem addressed.

4. Experimental Evaluation

This experiment aimed to train and test two CNN architectures, performing the analysis of their evaluation metrics in the task of classifying images of leaf diseases, evaluating the network's ability to identify and classify the diseases that can affect tomato leaves.

4.1. Development Environment

In this research, the adopted development environment was the Python language in version 3.11.3. For the implementation of the convolutional neural networks used and their callbacks, the Keras library was used [Chollet 2015]. In addition, Visualkeras [Gavrikov 2020] was used for the visualization of the implemented network architectures and the Matplotlib library [Hunter 2007] for graph plotting.

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 database used is composed of images of tomato leaves with different types of diseases. The tomato leaf images from the first dataset are selected from the PlantVillage database with ten categories, nine of which are intended for different foliar diseases.

Each image consists of a single leaf and a single background, the database contains a total of 14,531 images. The dataset was made available in JPEG format.

4.3. Preparation of Data and Classifiers

To mitigate overfitting in the limited database, data pre-processing techniques were employed. The proposed approach utilized Data Augmentation methods to expand the dataset size. Techniques such as rotation, inversion, and noise applied to each image. However, these transformations were exclusively implemented on the training set as recommended in Shorten and Khoshgoftaar [Shorten and Khoshgoftaar 2019]. Applying data augmentation to the entire dataset can lead to computational challenges and longer training times. To facilitate this process, the Albumentations [Buslaev et al. 2020] library was utilized.

The architectures were trained using 100 epochs, each epoch being defined as applying all images in the training set to the model. When an epoch is finished, the current state of the model is validated using the validation dataset.

The Early Stopping [1] callback was used. This is a common callback regularization technique used to avoid overfitting neural network models during training. The goal is to stop model training when performance on a validation set no longer improves.

In Keras, early stopping can be implemented using the EarlyStopping callback. This callback monitors a specified metric and halts the training process if there is no improvement in that metric for a specified number of epochs. In this study, the AUC metric was monitored, and a patience criterion of 5 epochs was set.

Finally, the Root Means Square Propagation optimizer was used, another hyperparameter used with the optimizer was the Loss function, CategoricalCrossentropy [2]. For

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

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

training, the value of the initial learning rate used was 1×10^{-3}, this rate was modified each time to a lower one.

5. Results and Discussions

DenseNet and Inception convolutional neural network models were trained and tested on a public database to classify diseases in tomato leaves. The models' performance was evaluated using different metrics, such as accuracy, AUC, precision, recall, and loss. The analysis showed that the DenseNet network outperformed the Inception network. In the test set, DenseNet achieved an accuracy of 98.25% and an AUC of 98.72%, while Inception had an accuracy of 93.02% and an AUC of 95.13%.

In Figure 3 can be seen the behavior of the DenseNet model metrics during the model training and validation process. It is verified that the model had its training interrupted with 30 epochs, due to the EarlyStopping callback.

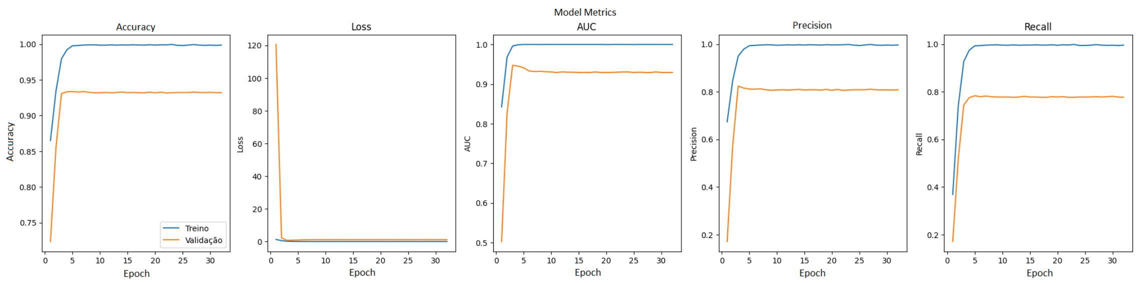

Figure 3. Results Obtained by the DenseNet Architecture

During the training process of the DenseNet model, using the validation set, after 23 epochs, the Loss function reached its minimum value (0.0164), and the corresponding accuracy was 99.86%, precision 99.59%, 99.56% recall and 100% AUC.

In Figure 4 we can see the behavior of the Inception model metrics during the model training and validation process. It is verified that the model had its training interrupted with 15 epochs, half of what was needed for the DenseNet model, due to the EarlyStopping callback.

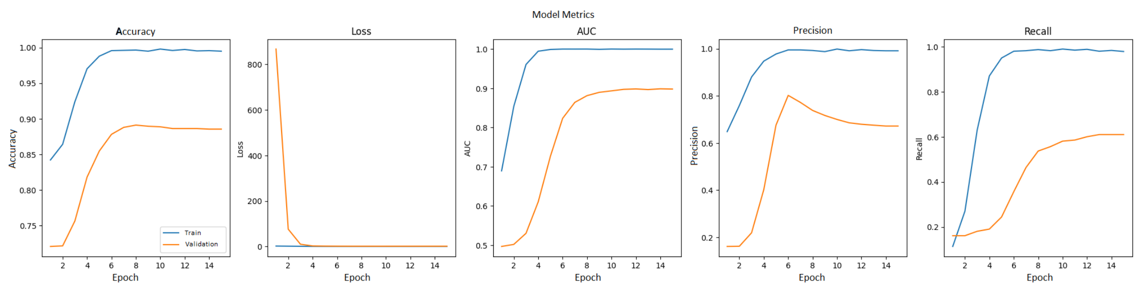

Figure 4. Results Obtained by the DenseNet Architecture

During the training process of the Inception model, using the validation set, after 12 epochs, the Loss function reached its minimum value (0.0568), and the corresponding accuracy was 99.74%, precision 99.64%, 98.80% recall and 99.98% AUC.

Although the Inception network was trained for only half the number of epochs compared to DenseNet, it still failed to outperform the metrics achieved by DenseNet. The Inception model showed lower results across all the evaluated metrics.

The Table 1 shows the behavior of the results obtained by the DenseNet architecture in the test set.

Table 1. DenseNet Architecture Results in the Test Set.

Metric	Value
Accuracy	98.25%
Precision	95.33%
Recall	94.09%
AUC	98.72%

The Table 2 shows the behavior of the results obtained by the Inception architecture in the test set.

Table 2. Inception Architecture Results in the Test Set.

Metric	Value
Accuracy	93.02%
Precision	82.20%
Recall	74.20%
AUC	95.13%

The results obtained by the DenseNet and Inception networks in the present work were evaluated in different performance metrics. DenseNet presented an accuracy of 98.25%, precision of 95.33%, recall of 94.09% and AUC of 98.72%. Inception, although it was interrupted with half the number of epochs required for the other network, obtained an accuracy of 93.02%, precision of 82.20%, recall of 74.21% and AUC of 95.13%.

Accuracy means that the model was right 98.25% of the time in classifying leaf diseases. It is an important metric, but it is important to remember that it can be misleading in some cases, when a class is much more frequent than the others, which can lead to a result with high accuracy, but with poor performance in the classification of classes minorities.

The precision achieved by the network means that when the model classifies a leaf as belonging to a given disease, it is correct approximately 95.33% of the time. Precision is important when you want to avoid false positives, that is, when the model classifies a leaf as sick, but it is healthy. This is especially important when the consequence of a false positive is serious, such as unnecessary intervention in the plantation.

The network recall means that the model is able to correctly identify 94.09% of leaves that actually belong to a given disease. Recall is important when you want to avoid false negatives, that is, when the model classifies a leaf as healthy when, in fact, it is sick. This is especially important when the consequence of a false negative is severe, such as letting a disease spread through the crop.

Finally, the AUC metric, with 98.72%, means that the model managed to achieve an excellent ability to distinguish between different leaf diseases and healthy leaves. The

AUC is an important metric because it considers the recall and specificity of the model, that is, its ability to correctly identify both positive and negative samples.

Through the analysis of the chosen metrics, it can be verified that the DenseNet network obtained a superior performance in the task of classifying diseases in tomato leaves, with an accuracy of 98.25 % and AUC of 98.72%, while the Inception network obtained an accuracy of 93.02% and AUC of 95.13%. The choice of the DenseNet network is due to the fact that this network showed better results in all evaluated metrics, including accuracy, precision, recall and AUC, indicating that this network is more efficient in classifying diseases in tomato leaves.

6. Conclusion and Future Work

In this study, two architectures of convolutional neural networks, DenseNet and Inception, were analyzed for the task of classifying foliar diseases that attack tomato. The dataset used was composed of a limited number of images, but it was optimized through pre-processing techniques, such as data augmentation, which significantly increased the number of images in the training set.

The results showed that the DenseNet architecture achieved an accuracy of 98.25%, while the Inception architecture achieved an accuracy of 93.02%. Although both models achieved satisfactory results, the DenseNet architecture stood out for its better generalization ability and superior performance in the task of classifying foliar diseases in tomato. The choice of the DenseNet network is due to the fact that this network has shown better results in all evaluated metrics, including accuracy, precision, recall and AUC, indicating that this network is more efficient in classifying diseases in tomato leaves.

One improvement identified in this work is using a single dataset for training and testing the classifier. To achieve a more generalizable model, we suggest incorporating additional databases during training. Furthermore, future research could explore alternative image pre-processing and data augmentation techniques to enhance the performance of convolutional neural networks in classifying plant leaf diseases. Developing more robust models is crucial in effectively monitoring and controlling diseases in agricultural crops, mitigating production losses, and ensuring food security.

References

Agarwal, M., Singh, A., Arjaria, S., Sinha, A., and Gupta, S. (2020). Toled: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science*, 167:293–301.

Bhattacharjee, D., Mandal, S., and Basak, J. (2019). Detection and classification of rice plant diseases using deep convolutional neural network. *Computers and Electronics in Agriculture*, 163:104853.

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. (2015). Keras: Deep learning library for theano and tensorflow. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 478–486.

Fuentes, A. and Yoon, S. (2018). Deep learning for plant disease recognition using mobilenet v2. *Sensors*, 18(12):4289.

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

Hong, H., Lin, J., and Huang, F. (2020). Tomato disease detection and classification by deep learning. In *2020 International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, pages 25–29. IEEE.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Huang, M.-L. and Chang, Y.-H. (2020). Dataset of tomato leaves.

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

Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:1419.

Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., and Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. In *2016 14th International Conference on Precision Agriculture*, pages 197–204. IEEE.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2018). Chestnet: A deep neural network for classification of thoracic diseases on chest radiography. *arXiv preprint arXiv:1807.03133*.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328.

Zaki, S. Z. M., Zulkifley, M. A., Stofa, M. M., Kamari, N. A. M., and Mohamed, N. A. (2020). Classification of tomato leaf diseases using mobilenet v2. *IAES International Journal of Artificial Intelligence*, 9(2):290.