Data and Metadata. 2023; 2:181
doi: 10.56294/dm2023181

ORIGINAL

Classification of diseases in tomato leaves with Deep Transfer Learning

Clasificación de enfermedades en hojas de tomate con Deep Transfer Learning

Noredine Hajraoui[1], Mourade Azrour[1], Ahmad El Allaoui[1]

[1]Moulay Ismail University Faculty of Science and Technology Errachidia. Morocco.

Cite as: Hajraoui N, Azrour M, El Allaoui A. Classification of diseases in tomato leaves with Deep Transfer Learning. Data and Metadata. 2023;2:181.https://doi.org/10.56294/dm2023181

Submitted: 07-08-2023 **Revised:** 16-09-2023 **Accepted:** 29-10-2023 **Published:** 01-12-2023

Editor: Javier Gonzalez-Argote
Guest Editor: Yousef Farhaoui

Note: paper presented at the International Conference on Artificial Intelligence and Smart Environments (ICAISE'2023).

ABSTRACT

Plant diseases are important factors because they significantly affect the quality, quantity, and yield of agricultural products. Therefore, it is important to detect and diagnose these diseases at an early stage. The overall objective of this study is to develop an acceptable deep learning model to correctly classify diseases on tomato leaves in RGB color images. To address this challenge, we use a new approach based on combining two deep learning models VGG16 and ResNet152v2 with transfer learning. The image dataset contains 55 000 images of tomato leaves in 5 different classes, 4 diseases and one healthy class. The results of our experiment are promising and encouraging, showing that the proposed model achieves 99,08 % accuracy in training, 97,66 % in validation, and 99,0234 % in testing.

Keywords: Tomato Diseases; Deep Learning; Transfer Learning.

RESUMEN

Las enfermedades de las plantas son factores importantes porque afectan significativamente a la calidad, la cantidad y el rendimiento de los productos agrícolas. Por lo tanto, es importante detectar y diagnosticar estas enfermedades en una etapa temprana. El objetivo general de este estudio es desarrollar un modelo de aprendizaje profundo aceptable para clasificar correctamente enfermedades en hojas de tomate en imágenes de color RGB. Para abordar este desafío, utilizamos un nuevo enfoque basado en la combinación de dos modelos de aprendizaje profundo VGG16 y ResNet152v2 con aprendizaje de transferencia. El conjunto de datos de imágenes contiene 55 000 imágenes de hojas de tomate en 5 clases diferentes, 4 enfermedades y una clase sana. Los resultados de nuestro experimento son prometedores y alentadores, mostrando que el modelo propuesto alcanza una precisión del 99,08 % en el entrenamiento, 97,66 % en la validación y 99,0234 % en las pruebas.

Palabras clave: Enfermedades del Tomate; Deep Learning; Transfer Learning.

INTRODUCTION

Tomato production in Morocco reached 1 409,44 million kilos in 2018 on an area of 15 955 hectares, with a yield of 8,83 kilos per square meter, making it one of the 15 largest tomato producing countries in the world, according to the statistics agency of the Food and Agriculture Organization of the United Nations (CAM: Food and Agriculture Organization). In the latest data (2018) on tomato production FAOSTAT 2018,[1] the statistics agency of the Food and Agriculture Organization of the United Nations (FAO), it was also noted that Morocco achieved one of the best yields with 8,83 kg per square meter, overtaking Spain and the Netherlands. Despite the efforts and development in tomato cultivation, Morocco suffers from a number of serious diseases that undermine the increase in crop yields and hinder progress. In addition, the use of tradition-al methods and the lack of experience of many farmers pose a major challenge. Correct disease detection is an early stage at which action can be taken to avoid losses and achieve high yield quality. Nowadays, leaf disease detection is a major concern for which visual applications are being developed and is the reason for the proliferation of digital technologies. Recently, the use of Deep Learning has shown good results in various fields, thanks to the large amount of data generated every day that can be used to train Deep models and the computational power provided by the built-in graphics processing units (GPUs). Thus, all researchers are taking advantage of the power and efficiency of Deep Learning in processing large amounts of data to find solutions for identifying and classifying of plant diseases, especially tomato leaves. Ouhami et al.[2] presented Deep Learning approaches with transfer learning. Three models were compared: VGG16 with an accuracy of 90,58 %, DensNet121 with an accuracy of 94,93 %, and DensNet161 with a result of 95,65 %. DensNet161 outperformed all classifiers. Marino et al.[3] attempt to locate and classify imperfections or defects in potatoes. They created a labeled dataset with 6 different classes and several variants. The images of potatoes were acquired using a multicamera device. A combination of autoencoders and SVMs was proposed to localize damaged areas and green areas in selected images. The localization results were used as input to the SVMs for classification. An average accuracy of 95 % and an average recognition rate of 93 % were obtained when classifying potato images into 6 classes. Brahimi et al.[4] tested several state-of-the-art CNN architectures for plant disease classification using a database of 55038 images removed and corrected from the PlantVillage public database. The database consists of 14 plant types divided into 39 classes of healthy and infected leaves, including a background class. The best results were obtained with the transfer learning model ResNet34, which achieved 99,67 % accuracy. Elhassouny et al.[5] proposed an intelligent mobile application based on a deep CNN model for leaf disease detection in tomato. The author used the MobileNet CNN model, which can detect 10 types of leaf diseases in tomato. The system was trained using 7176 images for disease detection by intelligent mobile systems. The proposed application works in real time. The model was optimized using various optimization algorithms such as adadelta optimizer, stochastic gradient descent, adagradDA, momentum, adam, proximolateral and RMSprop. The model can be extended to diagnose imperfections. To improve the accuracy of tomato disease detection, a set of good quality image samples is required. The proposed method achieves the highest accuracy of 90,03 % at a learning rate of 0,001. At learning rates of 0,01 and 0,05, the accuracy is 86,7 % and 88,9 %, respectively. Seth et al.[6] proposed a CNN architecture based on transfer learning for leaf disease classification in tomato. The dataset is a combination of all major datasets available online, such as Plant Village, Plant Doc, and Mandely, which include a total of 41 863 images divided into 10 classes ranging from health to different types of diseases of tomato leaves. The focus is on the ResNet50 network, a recognized CNN architecture that has the best detection accuracy of 99,2 % and is often compared with existing works. El Massi et al.[7] proposed a leaf disease detection method based on the combination of several SVM classifiers (sequential, parallel and hybrid) using color, texture and shape features. Several preprocessing steps are performed, including normalization, noise reduction, and segmentation by the Kmeans clustering method. The database of infected plant leaves contains six categories, including three pest threats and three disease symptoms. The hybrid method outperformed the other methods with an overall detection rate of 93,90 %. Tiwari et al.[8] use pre-trained models such as VGG19 for transfer learning to extract relevant features from the dataset. Then, results were obtained using various classifiers, including logistic regression. With SVM, the accuracy was 94,7 %, with KNN 95,4 %, with neural network 96,5 %, and with logistic regression 97,8 %. Kaggle is an open-source repository that provides a Plant Village dataset. Two types of images: RGB and grayscale. Mohanty et al.[9] used a transfer learning technique with a pre-trained AlexNet to classify plant diseases using the public PlantVillage dataset with 54306 images containing 38 classes of 14 plants and 26 diseases and achieved 99,35 % accuracy. Jasim et al.[10] present a leaf disease classification and detection system using Deep Learning techniques. The dataset is obtained from a website (Plant Village Dataset) and contains 20636 plant images divided into 15 classes, of which 12 represent diseases of different plants and 3 classes for healthy leaves. As a result, the proposed system based on convolutional neural network (CNN) has excellent accuracy in training and testing, so that the accuracy in training reaches 98,29 % and in testing reaches 98,029 %. El Massi et al.[11] present a method based on combining classifiers for automatic detection of leaf diseases/damage. The proposed system uses two combination variants: series (SC) and hybrid (HC). They are proposed to avoid the problem of similarity between classes (diseases/damages) based on their color. The first variant SC is a serial combination of two SVM classifiers S1 and S2. The second HC method is a hybrid combination of three SVM classifiers, which inclues a serial combination (similar to SC) of two classifiers H1 and H2 in parallel with an individual classifier H3 that uses texture and shape to discriminate between classes. In this study, the hybrid method can achieve the best results com-pared to the serial method, with the overall recognition rate of the hybrid combination being 91,11 % and that of the serial combination being 88,33 %. Francis et al.[12] developed a convolutional neural network for disease detection in apple and tomato plants.

The dataset used includes 3663 images. The model consists of four convolutional layers. Each convolutional layer is followed by a pooling layer. Two fully linked layers with a sigmoid function were used to detect disease probability. The model overfits because there was a gap between the validation curve and the learning curve. To solve the overfitting problem, the learning rate is set to 0,25. Tan et al.[13] tried to determine the most suitable ML /DL models for the PlantVillage dataset and the tomato disease detection problem. The performance of each ML /DL model varied from one dataset to another and from one problem to another. In this study, three machine learning algorithms (kNN, RF and SVM) and five deep learning algorithms (AlexNet, VGG16, ResNet34, EfficientNet-b0 and MobileNetV2) were tested for tomato disease classification. The results show that among the tested ML /DL algorithms, ResNet34 network is a good choice that meets the accuracy requirements of disease classification with 99,7 % accuracy, 99,6 % precision, 99,7 % recall, and 99,7 % F1 score. In Aravind et al.[14] tomato disease classification is based on the PlantVillage dataset images with a pre-trained deep learning architecture, namely AlexNet and VGG16 network. The classification accuracy for 13,262 images was 97,29 % for VGG16 and 97,49 % for AlexNet. Gangwar et al.[15] proposed a transfer learning approach for grape leaves using the InceptionV3 model with an SVM classifier, a neural network and logistic regression. In this study, a PlantVillage dataset with a total of 4062 data is used, which is composed of 4 classes, including 3209 data for training and 853 data for testing. The best classification results for InceptionV3 using the logistic regression classifier achieve 99,4 % accuracy. Habiba et al.[16] use a deep convolutional neural network for plant disease detection and classification in tomato. They used the VGG16 deep cnn classifier to detect unhealthy plants and their diseases on images of tomato plants. The Plant Village dataset was used, which contains ten different classes of tomato leaf images including healthy plants. Using the transfer learning method with the pre-trained VGG16 model, this dataset shows a satisfactory classification performance of about 95,5 %. The top-2 accuracy of this model reaches 99 % in detecting diseases in tomato plants. Without any segmentation or pre-processing of leaf images, the trained model shows about 100 % performance in discriminating healthy and diseased plants. The remainder of the paper is organized as follows: Section 2 de-scribes the proposed method and model used and the steps taken to obtain the required results. Section 3 deals with the results and evaluation of the proposed methodology. Section 4 contains the conclusions of the paper and gives an outlook on future work.

METHODS

Dataset and data preparation

The tomato image dataset used in this study is an open-source dataset containing a total of 5500 images derived from the Kaggle dataset and the image dataset published online at www. PlantVil-lage.org with over 50 000 images. The dataset is divided in a ratio of 80 % for training, 10 % for validation and 10 % for testing. The dataset is divided into 5 classes figure 1, 4 classes of tomato diseases besides the class of healthy tomatoes. These classes are Tomato_Bacterial_spot (TBS), Tomato_Early_blight (TEB), Tomato_Late_blight (TLB), Tomato_Leaf_Mold (TLM) and Toma-to_healthy (TH). Figure 2 shows the number of images for each class in the database. All images in the dataset were resized to 224 × 224×3 (RGB) so that they would be accepted as input to the models used and speed up the calculations. To compensate for the insufficiency of the existing data, the training and validation data were augmented.[17] using Keras via the ImageDataGenera-tor function, e.g., rotation translation,zoom, width and height shift, and horizontal flipping.

Figure 1. Some Images from the dataset

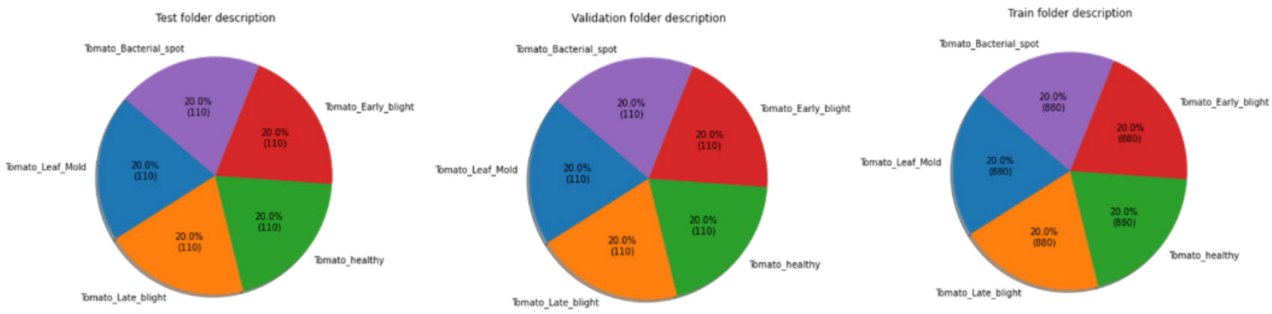

Figure 2. Dataset distribution

Motivation

The use of transfer learning for Deep Learning is based on the fact that knowledge previously learned from pre-trained models can be applied to another task to solve a new problem faster. Transfer learning also avoids the condition of needing a huge dataset and leverages the feature extraction capabilities of pre-trained models on data such as ImageNet.[18] This motivates us to use this technique in the proposed Deep Learning for tomato leaf disease classification.

Transfer Learning

With Transfer Learning, we can do Deep Learning without doing any computation. The principle is to use the knowledge acquired by a neural network in solving one problem to solve another problem that is more or less similar.[19] In this way, knowledge transfer is achieved. Transfer learning not only accelerates and optimizes the learning efficiency of the model,[20] but also helps to avoid overlearning. When the number of input images is small, it is not recommended to train the neural network to zero (i.e., with random initialization).[21] The reason is that the number of parameters to be learned is much higher than the number of images, and the risk of overfitting is enormous. Fine-tuning deep networks is a common method for transferring deep networks. It allows an already trained network to be used for a specific task such as classification with large amounts of data to overcome data insensitivity.[19] Fine-tuning saves time since the network does not need to be trained from scratch for new tasks.[21]

Pre-trained model architectures used

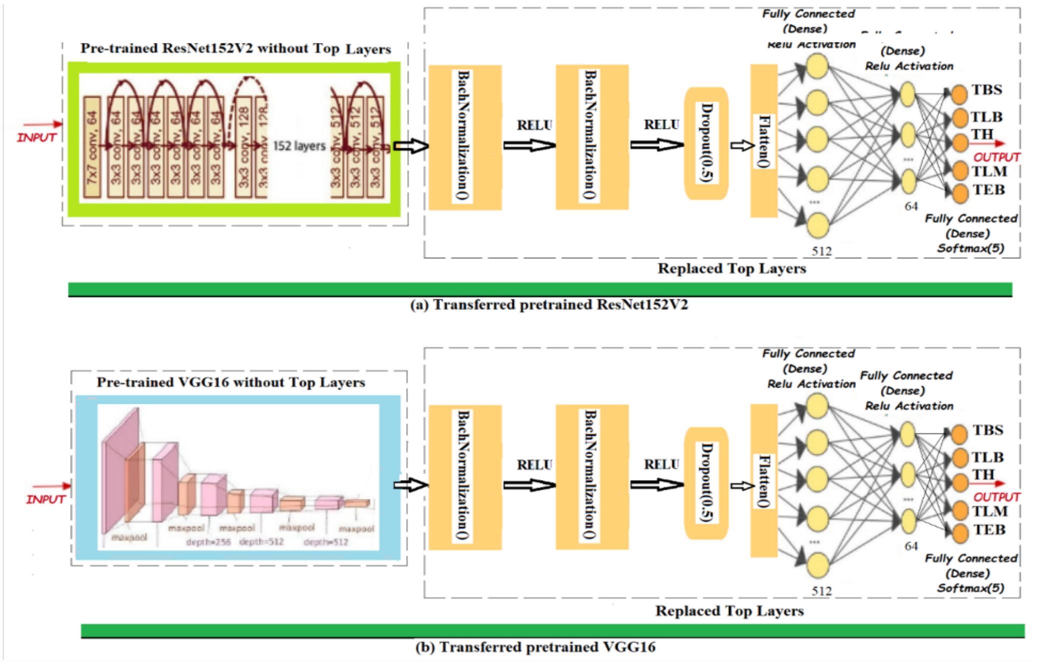

Figure 3. Architectures of modified deep transfer learning models (a) ResNet152V2, (b) VGG16

https://doi.org/10.56294/dm2023181

In this study, the transfer learning method uses the pre-trained models ResNet152V2 and VGG16. Both models were previously trained on the large ImageNet dataset. The models previously trained on the ImageNet data can help solve timing and computational problems as well as the lack of training data.

ResNet152v2 Transfer Learning

A CNN architecture with hundreds or thousands of convolutional layers is referred to as ResNet (Residual Network).[22] The efficiency of additional layers has been reduced by previous CNN configurations. ResNet has a large number of layers and is extremely fast. ResNetV2 applies batch normalization to each weighting layer. ResNet performs well on image recognition and localization tasks.[23] Figure 3 (a) shows the modified architecture of the Resnet152V2 model, whichconsists of 152 layers in depth and is mainly composed of 3-layer locks. In our study, for the transfer learning of Resnet152V2, the toplayer was removed and replaced with the following layers: two "BachNormalization" and "RELU" layers, one dropout layer, one flatten layer, and three dense layers, where the last dense layer uses the softmax classifierwith5classesasoutput.

TransferLearning VGG16

VGG was developed in 2014 by Simonyan and Zisserman of the Visual Geometry Group at the College of Oxford This model won second place in the 2014 ILSVRC competition[24] and consists of 16 chaws, 13 convolutional layers with a filter size 3x3, including five combined max-pooling layers that allow for a smaller volume size[24] and three fully connected layers. The final layer of the network is a softmax regression classifier that classifies input images based on prob-abilities. The input image size for VGG16 is fixed at 224 x 224 x 3. shows the modified architecture of the VGG16 model. We fine-tuned VGG16 by truncating the top 1000-class layer in the output and replacing it with our customs layers figure 3 (b). The number of our class labels is 5.

Proposed model

The proposed model for our classification problem was created by combining (concatenating) the two pre-trained models ResNet152V2 and VGG16. We applied these two predicted top layer models by transfer learning and fine tuning. Instead of the truncated layers, we added two consecutive "BachNormalization" layers, followed by a layer consisting of the activation function "ReLU", a dropout layer, a flatten layer, two fully connected (dense) layers with 512 neurons, "ReLU" and 64 neurons, "ReLU'.[25] Then we add the last dense layer with the SoftMax classifier as the activation function to build deep learning model that performs the five-class classification task figure 4.

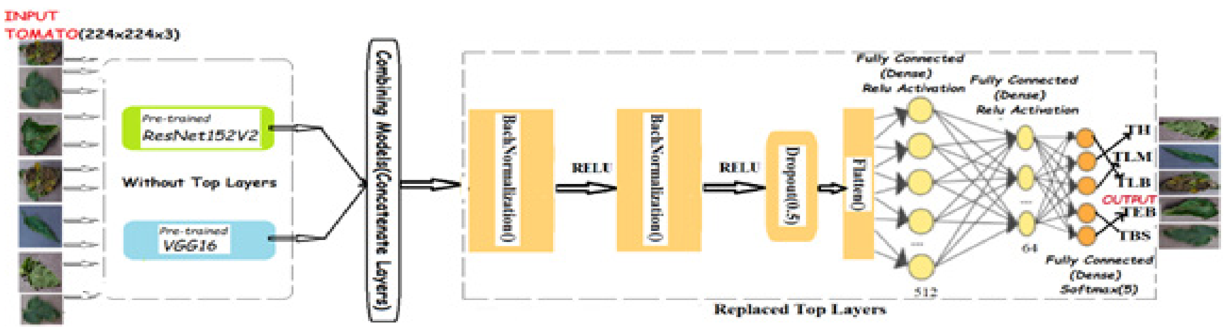

Figure 4. Proposed architecture deep transfer learning models

RESULTS AND DISCUSSION

Experimental design

Experiments were conducted online on Google Colaboratory (Colab)[26] using a 2.20 GHz Intel Xeon processor, 13 GB RAM, Tesla K80 GPU throttle, and a HP EliteBook 8570p computer with the following characteristics : Operating System : Windows 10, Operating System Architecture Type: 64-bit, an Intel(R) Core (TM) i7-3520M CPU @ 2.90GHz (4 CPUs), ~2.9GHz, 12 GB RAM and an AMD Radeon HD 7570M 4GB graphics card. The Python programming language was used to train the proposed deep transfer learning models, includingthe Keras package and a Tensorflow backend. Keras is an easy-to-use neural network library built on top of Theano or TensorFlow.[23] Keras provides most of the building blocks needed to create reason-ably sophisticated deep learning models. The model architecture is trained using an Adam optimizer[27] with a learning rate of 1e-4 and a total number of 175 epochs. The dataset is split into 80 % for training, 10 % for validation, and 10 % for evaluation (testing).

https://doi.org/10.56294/dm2023181

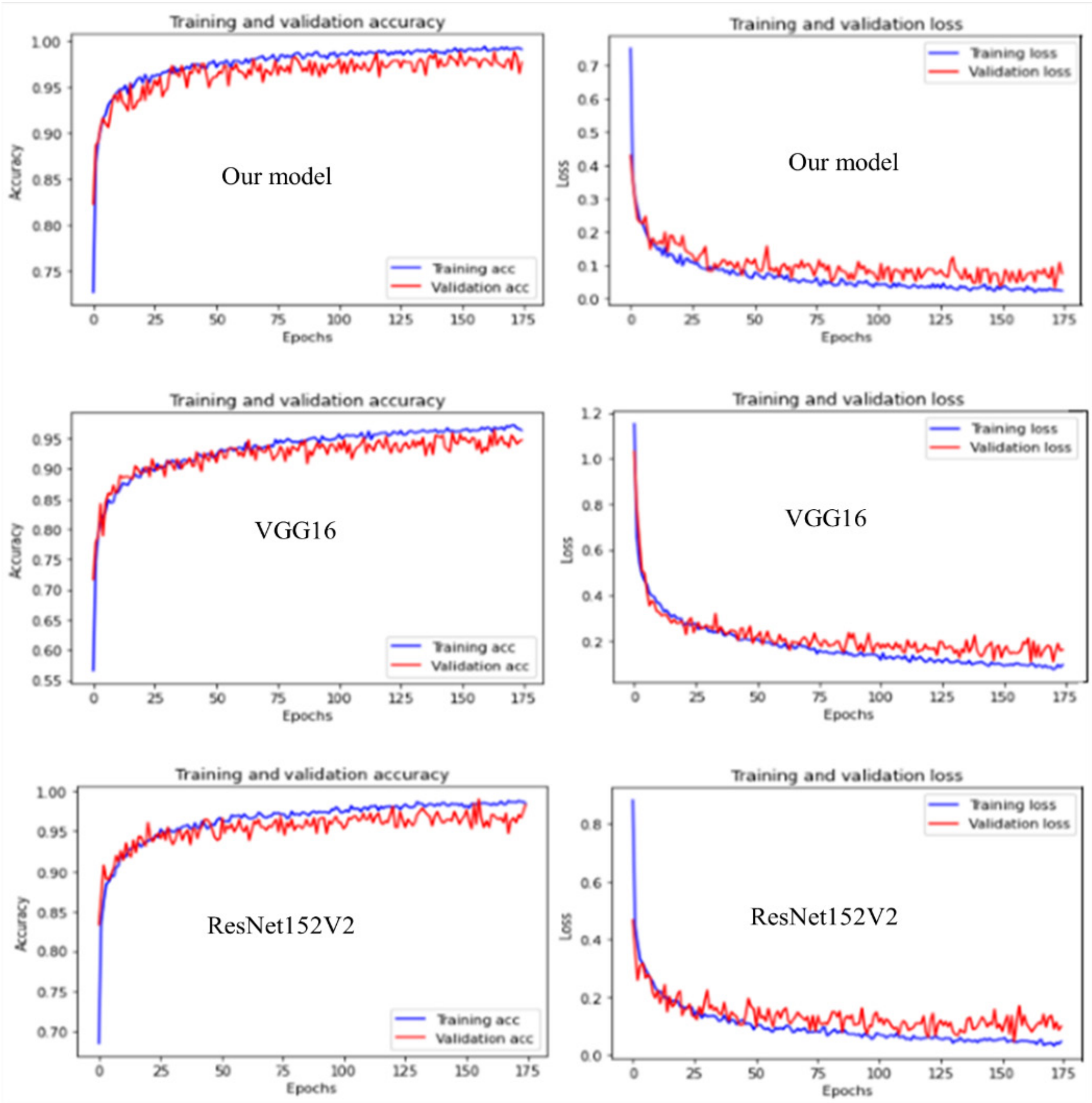

Figure 5. Accuracy and Loss for each model

Performance evaluation

The results of the experiments are shown in figure 5. Each figure shows the accuracy and loss values of the three proposed model architectures (ResNet152V2+VGG16), VGG16, and Res-Net152V2. After fine-tuning and over 175 epochs, all three models had accuracy greater than 94 %. During training iterations, high accuracy results were achieved with significantly reduced loss. Our model and ResNet152V2 performed better than VGG 16. They also converged slightly, as shown in figure 5. The deeper models achieved better test results, as shown in table 1. Our model increases its accuracy better and decreases its loss during iterations, as shown in figure 5. Overall, the proposed model performed well with the highest accuracy and lowest loss, followed by ResNet152V2, while VGG16 performed poorly with the lowest accuracy and highest loss. To compare and confirm the effectiveness of our proposed model, we calculated the confusion matrix of the test data set for these models. Figure 6 (a) (b) (c) shows the confusion matrix.

For our model, VGG16 and ResNet152V2. More images were misclassified for ResNet152V2 and VGG16 than for our proposed model. Moreover, the most confusing classes for VGG16 and ResNet152V2 are Tomato_Early_blight and Tomato_Late_blight for VGG16 and ResNet152V2 respectively.5 % and 1 % of Tomato_Early_blight images are classified as Tomato_Late_blight.7 % and 7 % of Tomato_Late_blight images are classified as Tomato_Early_blight for VGG16 and ResNet152V2 respectively. In the proposed model, the confusion is most likely between Toma-to_Early_blight and Tomato_Late_blight, so 4 % of Tomato_Late_blight images are classified as Tomato_Early_blight and 1 % of Tomato_Early_blight images are classified as Toma-to_Late_blight. The confusion matrices show that Tomato_Early_blight and Tomato_Late_blight are the most frequently misclassified diseases for the three models, which is due to the similarity between these two diseases that makes it difficult to distinguish between two classes. From table 1 and figures 5-6, our model performs better than the VGG16 and

ResNet152V2 models used individually. In the test phase, our model outperformed ResNet152V2 and VGG16 with 99,02 %, 97,07 % and 95,11 % accuracy, respectively. Figure 6 clearly shows that our model outperformed the other models in classifying Tomato_Bacterial_spot, Tomato_Early_blight, Toma-to_Late_blight, Tomato_Leaf_Mold and Tomato_healthy with 100 %, 99 %, 96 %, 100 % and 100 % accuracy, respectively.

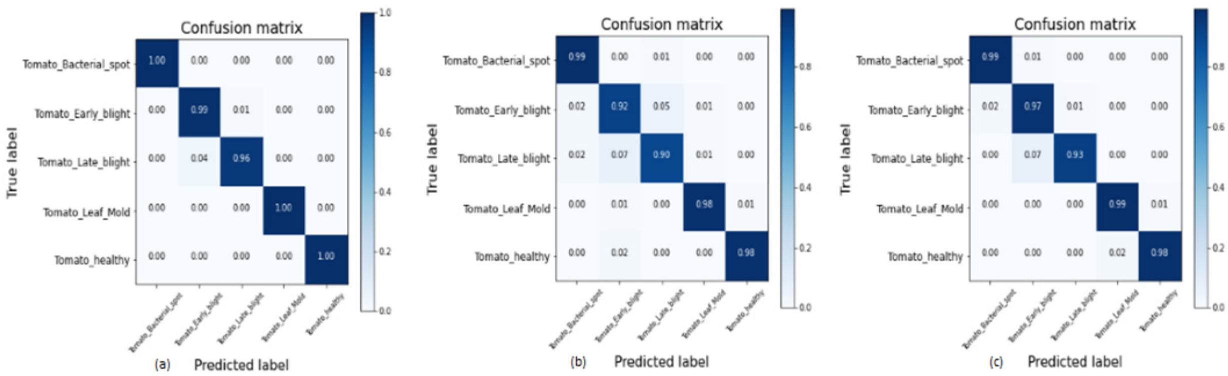

Figure 6. Confusion matrix of models: (a) Proposed model, (b) VGG16 & (c) ResNet152V2

Table 1. Accuracy and loss of training, validation, and testing for each model

Model	Training accuracy	Validation accuracy	Training loss	Validation loss	Test accuracy	Test loss
Our model	99,08 %	97,66 %	0,0242	0,0746	99,0234 %	0,035065
ResNet152V2	98,41 %	98,24 %	0,0464	0,0996	97,0703 %	0,102091
VGG16	96,38 %	94,73 %	0,0951	0,1604	95,1172 %	0,145772

Comparison of the state of the art

Table 2 describes the studies mentioned in section 1 that were compared with the results of our model. These approaches use a different data set. Based on the listed accuracies, it can be ob-served that almost all of these approaches are based on deep learning models, which shows the high accuracy compared to approaches based on other models. The results of our model are promising, starting from a dataset of 5500 images divided into 880 images for each class of the training phase and 110 images for each class of the validation and testing phase. They reach an accuracy of 99,02 % when combining the ResNet152V2 and VGG16 models.

Table 2. Comparison the State of Art

Authors	Crop	Data	Method	Accuracy
El Massi[8]	Tomato	600 images	SVM combination	93,90 %
Brahimi[5]	PlantVillage	55,038 images	Inception_v3	99,76 %
Ouhami[3]	Tomato	666 images	DensNet161	95,65 %
Elhassouny[6]	Tomato	7176 images	MobileNetet	90,03 %
Our model	Tomato	5500 images	ResNet152V2+VGG16	99,02 %

CONCLUSION

In this paper, we explore (two models) Deep Learning using the transfer learning method with the pre-trained models ResNet152V2 and VGG16 to build a model to solve a problem of leaf disease detection and classification in tomato. Our proposed ResNet152V2 and VGG16 model shows the highest test accuracy during 175 training periods, outperforming the tested architectures. The conducted research suggests that combining architectures is a good way to increase the accuracy of models suitable for plant disease detection based on plant images. The results are promising. In our future work, we will try to improve the results, increase the size of the dataset, extend our research with other pre-trained CNNs, and solve more difficult classification and disease detection problems and multi-classification tasks. In addition, we will apply our proposed model to more plants and diseases.

REFERENCES

1. FAOSTAT. 2018. Food and agriculture organization of the United Nations Database, Food and Agriculture

Organization Corporate Statistical Database.

2. M. Ouhami, Y. Es-Saady, M. El Hajji, A. Hafiane, R. Canals, and M. El Yassa, “Deep transfer learning models for tomato disease detection,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12119 LNCS, pp. 65–73, 2020, doi: 10.1007/978-3-030-51935-3_7.

3. S. Marino, P. Beauseroy, and A. Smolarz, “Deep learning-based method for classifying and localizing potato blemishes,” ICPRAM 2019 - Proc. 8th Int. Conf. Pattern Recognit. Appl. Methods, no. Icpram, pp. 107–117, 2019, doi: 10.5220/0007350101070117.

4. Mohammed Brahimi, Deep learning for plants diseases. Springer International Publishing, 2018. doi: 10.1007/978-3-319-90403-0.

5. Elhassouny and F. Smarandache, “Smart mobile application to recognize tomato leaf diseas-es using Convolutional Neural Networks,” Proc. 2019 Int. Conf. Comput. Sci. Renew. Ener-gies, ICCSRE 2019, pp. 1–4, 2019, doi: 10.1109/ICCSRE.2019.8807737.

6. V. Seth, R. Paulus, M. Kumar, and A. Kumar, Tomato Leaf Diseases Detection, vol. 894 LNEE. 2022. doi: 10.1007/978-981-19-1677-9_5.

7. El Massi, I., Es-Saady, Y., El Yassa, M., Mammass, D., Benazoun, “A hybrid combina-tion of multiple SVM classifiers for automatic recognition of the damages and symptoms on plant leaves,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9680, pp. 40–50, 2016, doi: 10.1007/978-3-319-33618-3_5.

8. D. Tiwari, M. Ashish, N. Gangwar, A. Sharma, S. Patel, and S. Bhardwaj, “Potato Leaf Dis-eases Detection Using Deep Learning,” in Proceedings of the International Conference on In-telligent Computing and Control Systems, ICICCS 2020, May 2020, pp. 461–466. doi: 10.1109/ICICCS48265.2020.9121067.

9. S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant dis-ease detection,” Front. Plant Sci., vol. 7, no. September, pp. 1–10, 2016, doi: 10.3389/fpls.2016.01419.

10. M. A. Jasim and J. M. Al-Tuwaijari, “Plant Leaf Diseases Detection and Classification Using Image Processing and Deep Learning Techniques,” Proc. 2020 Int. Conf. Comput. Sci. Softw. Eng. CSASE 2020, pp. 259–265, 2020, doi: 10.1109/CSASE48920.2020.9142097.

11. El Massi, Y. Es-saady, M. El Yassa, and D. Mammass, “Combination of multiple classifi-ers for automatic recognition of diseases and damages on plant leaves,” Signal, Image Video Process., vol. 15, no. 4, pp. 789–796, 2021, doi: 10.1007/s11760-020-01797-y.

12. M. Francis and C. Deisy, “Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks - A Visual Understanding,” 2019 6th Int. Conf. Signal Pro-cess. Integr. Networks, SPIN 2019, pp. 1063–1068, 2019, doi: 10.1109/SPIN.2019.8711701.

13. Tan, J. Lu, and H. Jiang, “Tomato Leaf Diseases Classification Based on Leaf Images: A Comparison between Classical Machine Learning and Deep Learning Methods,” AgriEngi-neering, vol. 3, no. 3, pp. 542–558, 2021, doi: 10.3390/agriengineering3030035.

14. Romero-Carazas R. Prompt lawyer: a challenge in the face of the integration of artificial intelligence and law. Gamification and Augmented Reality 2023;1:7–7. https://doi.org/10.56294/gr20237.

15. Gonzalez-Argote J. A Bibliometric Analysis of the Studies in Modeling and Simulation: Insights from Scopus. Gamification and Augmented Reality 2023;1:5–5. https://doi.org/10.56294/gr20235.

16. Gonzalez-Argote D, Gonzalez-Argote J, Machuca-Contreras F. Blockchain in the health sector: a systematic literature review of success cases. Gamification and Augmented Reality 2023;1:6–6. https://doi.org/10.56294/gr20236.

17. R. Aravind Krishnaswamy, R. Purushothaman, and A. Ramesh, “Tomato crop disease classi-fication using

pre-trained deep learning algorithm," Procedia Comput. Sci., vol. 133, pp. 1040–1047, 2018, doi: 10.1016/j.procs.2018.07.070.

18. Gangwar, D. Tiwari, A. Sharma, M. Ashish, A. Mittal, and G. K. Vishwavidyalaya, "Grape Leaf Diseases Classification using Transfer Learning," no. September, pp. 3171–3177, 2020.

19. S. U. Habiba and M. K. Islam, "Tomato Plant Diseases Classification Using Deep Learning Based Classifier from Leaves Images," 2021 Int. Conf. Inf. Commun. Technol. Sustain. Dev. ICICT4SD 2021 - Proc., pp. 82–86, 2021, doi: 10.1109/ICICT4SD50815.2021.9396883.

20. L. Perez and J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," 2017, [Online]. Available: http://arxiv.org/abs/1712.04621

21. L. Fei-Fei, J. Deng, and K. Li, "ImageNet: Constructing a large-scale image database," J. Vis., vol. 9, no. 8, pp. 1037–1037, 2010, doi: 10.1167/9.8.1037.

22. J. Pan and Q. Yang, "A Survey on Transfer Learning - {IEEE} Xplore Document," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2009.

23. Hu, J. Lu, and Y. P. Tan, "Deep transfer metric learning," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 07-12-June, pp. 325–333, 2015, doi: 10.1109/CVPR.2015.7298629.

24. H. Lei et al., "A deeply supervised residual network for HEp-2 cell classification via cross-modal transfer learning," Pattern Recognit., vol. 79, pp. 290–302, 2018, doi: 10.1016/j.patcog.2018.02.006.

25. Gulli and S. Pal, "Deep Learning with Keras," p. 318, 2017, [Online]. Available: https://www.packtpub.com/big-data-and-business-intelligence/deep-learning-keras

26. Kumar, H. Arora, Harsh, and J. Sisodia, "ResNet-based approach for Detection and Classi-fication of Plant Leaf Diseases," Proc. Int. Conf. Electron. Sustain. Commun. Syst. ICESC 2020, no. Icesc, pp. 495–502, 2020, doi: 10.1109/ICESC48915.2020.9155585.

27. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

28. J. Brown, L. A. Hutchinson, M. J. Rainbow, K. J. Deluzio, and A. R. De Asha, "A compar-ison of self-selected walking speeds and walking speed variability when data are collected during repeated discrete trials and during continuous walking," J. Appl. Biomech., vol. 33, no. 5, pp. 384–387, 2017, doi: 10.1123/jab.2016-0355.

29. T. Carneiro, R. V. M. Da Nobrega, T. Nepomuceno, G. Bin Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," IEEE Access, vol. 6, pp. 61677–61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

30. D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

FINANCING

The authors did not receive financing for the development of this research.

CONFLICT OF INTEREST

"The authors declare that there is no conflict of interest.

AUTHORSHIP CONTRIBUTION

Conceptualization: Noredine Hajraoui, Mourade Azrour, Ahmad El Allaoui.
Research: Noredine Hajraoui, Mourade Azrour, Ahmad El Allaoui.
Drafting - original draft: Noredine Hajraoui, Mourade Azrour, Ahmad El Allaoui.
Writing - proofreading and editing: Noredine Hajraoui, Mourade Azrour, Ahmad El Allaoui.

https://doi.org/10.56294/dm2023181