Detection of Plant Diseases Using Leaf Images and Machine Learning

Conference Paper · March 2022

DOI: 10.1109/INFOTEH53737.2022.9751245

4 authors:

Almira Suljovic
University of Donja Gorica
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Stevan Cakic
University of Donja Gorica
22 PUBLICATIONS **117** CITATIONS
SEE PROFILE

Tomo Popovic
University of Donja Gorica
167 PUBLICATIONS **1,269** CITATIONS
SEE PROFILE

Stevan Šandi
University of Donja Gorica
18 PUBLICATIONS **195** CITATIONS
SEE PROFILE

Detection of Plant Diseases Using Leaf Images and Machine Learning

Almira Suljović
Faculty of Applied Sciences
University of Donja Gorica
Podgorica, Montenegro
almira.suljovic@udg.edu.me

Stevan Čakić, Tomo Popović, *Senior IEEE Member*, Stevan Šandi
Faculty for Information Systems and Technologies
University of Donja Gorica
Podgorica, Montenegro

***Abstract*— Prevention and early detection of plant diseases is one of the main issues and challenges in agriculture. Farmers spend a lot of time observing and detecting diseased plants, often by looking at and analyzing plant leaves. Inadequate handling of plant disease such as late detection or the use of wrong pesticides often causes damage to crops, which causes a deterioration in the quality of food. This problem could be addressed using artificial intelligence and machine learning to detect plant diseases by processing digital images of leaves. As the leaf is the best indicator of whether the plant is healthy or not, by applying machine learning we can create predication models to detect the condition of the leaf in a shorter period of time and possibly prevent or reduce the losses. This paper describes experimenting with Detectron2 software library and Faster R-CNN neural network in order to detect the condition of the leaf. A dataset containing 6407 images was used to train the model. The original dataset has been extended by augmenting images using the RoboFlow tool. The experimentation and implementation was done using Google Colab, environment designed for cloud computing and machine learning development.**

Keywords*— *artificial intelligence; image processing, leaf disease detection; machine learning; plant disease detection; precision agriculture

I. INTRODUCTION

Artificial intelligence (AI) was recognized as a scientific discipline in the 1950s, but in its early years, due to insufficient technology and computing performance, it has not been widely explored [1]. As part of computer science, AI involves many activities such as learning, understanding, reasoning and interaction, with an aim to make a computer behave and "think" like a human being. There are several domains and fields withing AI, but one of the most important is machine learning, which led to the redevelopment of AI after the "winter of artificial intelligence" [2]. In recent years, there is an important development of machine learning called deep learning in which the algorithms use static techniques to develop problem solving models [2]. Machine learning has advanced greatly in the last two decades, and it is finding its ways into commercial use. A variety of machine learning algorithms provides the ability to select the appropriate problem based on the type of data and the type of problem being solved. The use of machine learning algorithms relies on finding a solution based on well defined input-output pairs in order to return the get the prediction model, a trained solution, that will provide the appropriate output data, even when presented with the inputs not used in the learning process [3].

Even at the beginning of the 20th century, agriculture production had low productivity. It could feed the population, but much larger farms and farmers were needed. The modernization of agriculture resulted in the use of nitrogen, synthetic pesticides, fertilizers, and better machinery, and the potential and scope of work increased [4]. With introduction of computers and various digital tools, smart agriculture is transforming the agricultural sector in terms of economic, social and environmental sustainability. Digitalization in agriculture leads to more effective production using resources as efficiently as possible, while targeting better quality of the product. The development of agriculture and smart development of agriculture can help attract new breeders [5].

Plant diseases pose a threat to food security in the world of trade, and they can have catastrophic coincidences for farmers whose lives depend on the success of production [6]. Plant disease are considered a serious problem for farmers because they can cause large economic losses, crop loss, and early detection of the diseases allows for an early response and rescue of crops [7]. Crop loss due to pests and diseases that attack plants can be more than 50% [6]. Timely detection of plant diseases is a very important and challenging task for farmers, which can be facilitated with advanced digital tools [8]. Often, plant leaves are indicators whether the plant is healthy or not. In most cases, it is necessary to have farmers and/or agronomist to visually detect a possible disease and sometimes a laboratory test is needed, as well. With the development of AI, it is becoming possible to support this process with the help of image processing and machine learning [7].

There is an active research in this area. For example, using the Stacked Siamese Matching (SSM) network architecture to identify plant diseases with a small dataset with obtained accuracy of 92,7% on one dataset, while on the other dataset obtained accuracy was 94,3% [9]. Also, using convolutional neural network (CNN), with a dataset of 92206 images, to identify plant diseases, the resulting prediction model achieves

This research was funded in part by the EUROCC project, European High-Performance Computing Joint Undertaking (JU) under grant agreement No. 951732. The JU receives support from the EU's Horizon 2020 research and innovation programme and EUROCC project participating institutions.

an overall classification accuracy of 94% [10]. Another example is the use of CNN to detect potato diseases on a leaf dataset, the resulting model achieves an accuracy of 93.48% [11]. Leaf detection and classification can be performed using You Only Look Once (YOLOv3) object detector and ResNet18 transfer learning models. Using these tools, a model with an accuracy of 78% was obtained [12].

In this paper, the focus is on developing a prediction model for detecting if a leaf is healthy or unhealthy using Detectron2 and the dataset used for this research contains various plant species including apple, bell pepper, blueberry, cherry, corn, grape, peach, potato, raspberry, soybean, strawberry and tomato.

II. Materials and Methods

A. The Dataset: Plant Leaf Images

The dataset used for this research is provided by Singh et al. at the Indian Institute of Technology and is openly available on RoboFlow [13]. The dataset contains 6407 images of plant leaf in JPEG file format. It is divided into three folders, designated as *train*, *valid* and *test*, that contain images to be used for training, validation and testing. The original folder dataset has 2569 images and 30 classes for image classification [13]. Using Roboflow tool, the dataset was reorganized into two classes that contain images with leaves annotated as healthy or unhealthy. As a next step, an 80/10/10 split of the dataset has been performed: 80% is used for training, 10% of images is used for validation, and 10% for testing. Therefore, for the experiments in the research, the *valid* folder contains 641 images, *test* folder 640 images, and *train* folder 5126 images. Each of these three folders contains corresponding Common Object in Context (COCO) annotation files. These files represent annotations for each image.

The COCO dataset is formatted in JSON and has information sections such as info, licenses, images, annotations, categories and segment info for each image. Info section contains information about the dataset, licenses section contains a list of image licenses from the dataset. The image section contains list of images in dataset without labels, bounding boxes, of segmentation specified it only contains images and information about each one. COCO has more annotation types. For this research, the annotation type used was object detection, while remaining types are keypoint detection, stuff segmentation, panoptic segmentation and image captioning [14]. Figure 1 shows a sample image of plant leaf and the same image with corresponding annotations.

B. Pre-processing of Images: the Use of Roboflow

It is often necessary, in order for the model to be successfully trained, to make some changes with the training dataset. The original dataset can be downloaded with images in two different dimensions. One is 100x100 pixels and the other 416x416 pixels. For this study, 416x416 pixels images were taken and resized to 200x200 pixels. The original dataset has 30 classes in which the number of images is not the same, which means that the dataset is not ideal for training models. In order to enable better model training from the original dataset that contains 30 classes, we reorganized it so that the images were divided into two classes that represent healthy and unhealthy leaves. For example, the original annotations such as apple leaf and apple scab leaf, were renamed into annotations as leaf healthy and leaf unhealthy.

For the purpose of the dataset preparations, the online tool called RoboFlow was used. The features provided by RoboFlow are the division of the dataset into the appropriate train, validation and test parts. Furthermore, with the help of this tool it is possible to perform preprocessing, as well as augmentation of the dataset in order to increase the number of images in the dataset by several times more. RoboFlow provides the ability to annotate data, check the health of datasets, merge datasets as well as image transformation [15].

As previously mentioned, in order to improve the dataset, artificial data were created by zooming, asymmetrically cropping or rotating existing images. On the images from the original dataset, a rotation clockwise, counter-clockwise and upside down, and saturation between -25% and +25% were performed, which resulted in a larger, augmented training dataset.

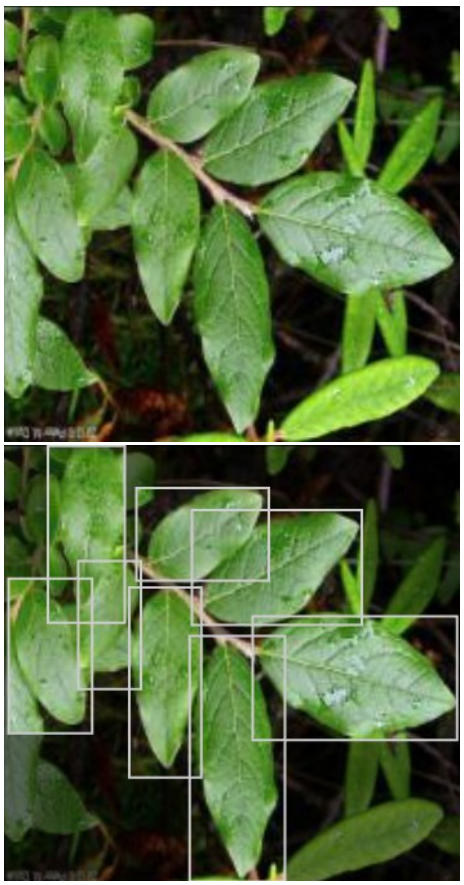

Figure 1. An image example from the dataset: an image of leaves without annotation (top), an image with annotations (bottom)

C. Tools Selection

The main tools used in this experiment were: Google Colab, Detectron2, Tensorboard, and programming was done in Python. The training, validation and testing of the model were performed on a free version of Google Colab. Google Colab is based on the Jupyter Notebook and designed as an

environment for machine learning and research education. Colaboratory notebooks can be shared with other users. Colab provides pre-configured Python 2 and 3 runtime with basic machine learning and artificial intelligence libraries, such as TensorFlow, Matplotlib and Keras. After the runtime, the virtual machine environment expires, it is deactivated and all user data and configurations are deleted, but it is possible to save the user data and configurations to a Google Drive account. Google Colab also provides GPU uptime, and is hosted on the Google Cloud platform [16].

For the implementation, Detectron2 software library was used, which is multi-purpose library that provides functions and algorithms for object detection. Detectron2 includes many object detection models and aims to perform high-speed training. For this application, Detectron2 uses the convolutional neural network (CNN) model called Faster region-based convolutional neural network (R-CNN), which is based on Fast R-CNN [17]. With R-CNN, it is necessary to send the region suggestion while with Fast R-CNN, an input image is sent to generate a convolutional feature map. From the convolutional feature map, the regions of the proposal are identified and folded into squares that are reshaped to a fixed size using the region of interest (RoI) polling layer, and can be fed into a fully connected layer. From the feature vector, RoI, we use softmax layer to predict the class of the proposed region, as well as the offset values for the bounding box. The Fast R-CNN is faster than the R-CNN because it is necessary to perform the convolution operation only once per image, and a feature map is generated from it [17]. Faster R-CNN is composed of Fast R-CNN and deep fully convolutional network that proposes regions, and is used for object detection [18]. The motivation for using Detectron2 and Faster R-CNN comes from the fact that Detectron2 includes high-quality object detection implementations and Faster R-CNN is faster than its predecessors. Therefore, it can be used for real-time object detection.

III. Results and Discussion

Figure 2. illustrates one approach to implement the system based on mobile app. The first step for implementation is to build an inference model. As shown for building an inference model, there are several steps such as selection of the dataset, preprocessing using RoboFlow, then the utilization of Detectron2 library with Faster R-CNN for the prediction model training. After the training, the resulting ML inference model is incorporated in the back-end functionality of the system and made available to farmers via mobile app. The farmer can take an image with the mobile phone, the image is sent to the back-end, and the health status can be returned to the farmer within the same app. In addition, the data can be collected and visualized for interaction with agronomists and other remote users.

During the study, while developing the prediction model, several runs of the training were executed. We experimented with different settings of the hyperparameters for training, and the following set of parameters was identified to yield the best results:

- Number of workers = 2, represents number of parallel data loading workers.
- Images per batch = 4, represents how many images contains one batch.
- Base learning rate = 0.001, represents learning rate for every group by default.
- Warm up iterations = 100, represents number of iterations for model warming up, parameter which is used to lower the learning rate in order to reduce the impact of deviating the model.
- Maximum iterations = 1600, represents the maximum number of iterations for training model.
- Gamma = 0.2, gamma is parameter that decrease the iteration number to learning rate.
- Batch size per image = 256.
- Number of classes = 3, in dataset we have two classes, healthy and unhealthy leaf, but when we preprocess the data with RoboFlow, it automatically adds a class that represents class for all images [19].

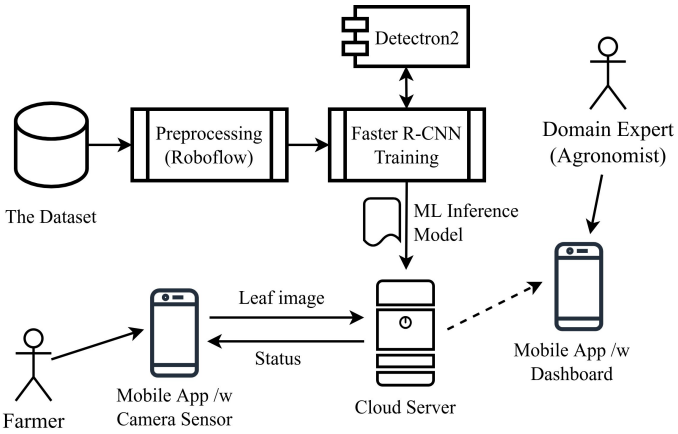

Figure 2. Implementation approach for the proposed solution

The evaluation of the results can be done by analyzing the various parameter such as training loss and average precision. For selected parameters, the progress of training loss over 1600 iterations and validation loss from 400th iteration is shown in Fig. 3. It can be seen that the training loss is getting better for last iterations. Each run of the training session to obtain the prediction model would take up to two and a half hours on the Google Colab GPU.

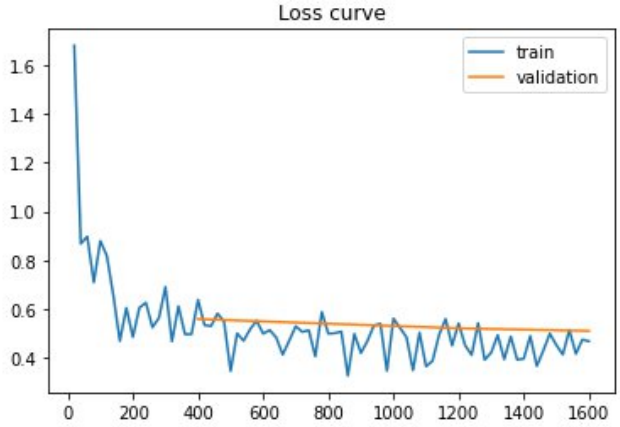

Figure 3. Training and validation loss

Figure 4 shows average precision parameters which were obtained for this model. These parameters are used for characterizing the performance of an object detector. The AP is average precision, mean average precision, for all 10 Intersection over Union (IoU) and all categories. AP50 metric is average over multiple IoU with the IoU threshold of 0.50. For other AP metrics difference is the IoU threshold [14]. For this model AP50 is 64.193, which indicates that this model is solid in detecting healthy and unhealthy leaves, but the results can be improved. The result obtained with Faster R-CNN residual neural network (ResNet) on similar dataset was reported as AP = 38.9 [13], close to our result of 39.852.

AP	AP50	AP75	APs	APm	APl
39.852	64.193	43.070	13.630	40.537	64.419

Figure 4. Average precession parameters achieved

The testing of the prediction model as also conducted on non-dataset data. An example input and output of these tests is illustrated in Fig. 5. The figure shows the input to the model, and the corresponding output providing the rectangle around each leaf detected and the assessment if the leaf is recognized as healthy or not.

Figure 5. An example image presented to the test model (left); the output annotations generated by the model under test (right)

IV. CONCLUSIONS

This paper describes research efforts on the use of machine learning, namely the Faster-R-CNN learning algorithm in order to develop prediction models for plant disease detection based on leaf image processing. The implementation was based on the use of Detectron2 software library using Python language and Google Colab environment for scientific computing. Initial experiments show promising result, but more research is needed.

Even through the obtained inference model provides satisfactory results, there is still much work to be done. In order to build a good and reliable detection model, it is important to collect more data that cover all types of leaves needed for its practical usage.

Future research steps will include experimenting with various preprocessing steps and augmentation techniques in order to enhance the dataset, the inclusion of additional datasets, and the use of different training models and configurations. We will also explore the use of HPC resources instead of free Colab environment.

REFERENCES

[1] M. Haenlein and A. Kaplan, “A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence,” California Management Review, vol. 61, no. 4, pp. 5–14, Jul. 2019, doi: 10.1177/0008125619864925.

[2] Daniel Castro, Joshua New, “The promise of artificial intelligence,” 2016.

[3] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” Science, vol. 349, no. 6245, 2015.

[4] I. Kovács and I. Husti, “The role of digitalization in the agricultural 4.0 – how to connect the industry 4.0 to agriculture?,” Hung. Agric. Eng., no. 33, 2018.

[5] A. M. Ciruela-Lorenzo, A. R. Del-Aguila-Obra, A. Padilla-Meléndez, and J. J. Plaza-Angulo, “Digitalization of agri-cooperatives in the smart agriculture context. Proposal of a digital diagnosis tool,” Sustainability, vol. 12, no. 4, 2020.

[6] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” Front. Plant Sci., vol. 7, 2016.

[7] A. Smetanin, A. Uzhinskiy, G. Ososkov, P. Goncharov, and A. Nechaevskiy, “Deep learning methods for the plant disease detection platform,” in proceedings of the 24th international scientific conference of young scientists and specialists (AYSS-2020), 2021.

[8] S. D. Khirade and A. B. Patil, “Plant disease detection using image processing,” in 2015 International Conference on Computing Communication Control and Automation, 2015.

[9] S. Jadon, “SSM-net for plants disease identification in low data regime,” in 2020 IEEE / ITU International Conference on Artificial Intelligence for Good (AI4G), 2020.

[10] A. A. Ahmed and G. H. Reddy, “A mobile-based system for detecting plant leaf diseases using deep learning”, AgriEngineering, vol. 3, no. 3, 2021.

[11] V. L. M. Dârdală, “Detecting plant diseases using deep learning architectures”, 2020

[12] Aravindhan Venkataramanan, Deepak Kumar P Honakeri, Pooja Agarwal, Ed., “Plant disease detection and classification using deep neural networks”, vol. 11, no. 0975–3397. International Journal on Computer Science and Engineering, 2019.

[13] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, “PlantDoc: A dataset for visual plant disease detection”, arXiv [cs.CV], 2019.

[14] “COCO - common objects in context”, Cocodataset.org. [Online]. Available: https://cocodataset.org/ [Accessed: 09-Dec-2021].

[15] “Overview”, Roboflow.com. [Online]. Available: https://docs.roboflow.com [Accessed: 09-Dec-2021].

[16] T. Carneiro, R. V. Medeiros Da Nobrega, T. Nepomuceno, G.-B. 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, 2018.

[17] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks”, arXiv [cs.CV], 2015.

[18] A. Shrivastava and A. Gupta, “Contextual priming and feedback for faster R-CNN”, in Computer Vision – ECCV 2016, Cham: Springer International Publishing, 2016.

[19] “Detectron2.Config – detectron2 0.6 documentation”, Readthedocs.io. [Online]. Available: https://detectron2.readthedocs.io/en/latest/modules/config.html [Accessed: 09-Dec-2021].