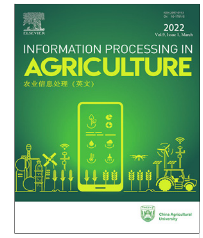
Available at www.sciencedirect.com

INFORMATION PROCESSING IN AGRICULTURE 9 (2022) 38–47

journal homepage: www.elsevier.com/locate/inpa



An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning

José G.M. Esgario [a], *Pedro B.C. de Castro* [a], *Lucas M. Tassis* [a], *Renato A. Krohling* [b,*]

[a] *LABCIN - Nature Inspired Computing, Federal University of Espirito Santo, Av. Fernando Ferrari, 514, CEP 29075-910 Vitória, Espírito Santo, ES, Brazil*

[b] *LABCIN - Nature Inspired Computing, and PPGI - Graduate Program in Computer Science, and Production Engineering Department, Federal University of Espirito Santo, Av. Fernando Ferrari, 514, CEP 29075-910 Vitória, Espírito Santo, ES, Brazil*

A R T I C L E I N F O

Article history:
Received 1 May 2020
Received in revised form
23 December 2020
Accepted 18 January 2021
Available online 1 February 2021

Keywords:
Convolutional neural networks
Semantic segmentation
Coffee leaf
Pests and diseases
App. development

A B S T R A C T

In recent years, deep learning methods have been introduced for segmentation and classification of leaf lesions caused by pests and pathogens. Among the commonly used approaches, convolutional neural networks have provided results with high accuracy. The purpose of this work is to present an effective and practical system capable of segmenting and classifying different types of leaf lesions and estimating the severity of stress caused by biotic agents in coffee leaves using convolutional neural networks. The proposed approach consists of two stages: a semantic segmentation stage with severity calculation and a symptom lesion classification stage. Each stage was tested separately, highlighting the positive and negative points of each one. We obtained very good results for the severity estimation, suggesting that the model can estimate severity values very close to the real values. For the biotic stress classification, the accuracy rates were greater than 97%. Due to the promising results obtained, an App for Android platform was developed and implemented, consisting of semantic segmentation and severity calculation, as well as symptom classification to assist both specialists and farmers to identify and quantify biotic stresses using images of coffee leaves acquired by smartphone.

1. Introduction

Plants are constantly exposed to a wide variety of biotic agents such as pests and pathogens (insects, viruses, fungi, bacteria, etc.) and also by abiotic factors such as water deficit, heat, salinity, and cold [15]. In this context, it is important to consider that plant diseases, caused by pathogens, lead to an impairment of the normal state of a plant, interrupting and modifying its vital functions. They can be a limiting factor in productivity and lead to significant crop losses [18].

Many measures can be adopted to prevent the spread of pests and diseases in plantations. In this sense, the integrated pests and diseases management reduces the chances of crop losses and reduce the need to use pesticides. For efficient pests and diseases management it is important not only to diagnose but also to quantify plant stress, as both functions are equally important for phytopathology [7].

According to the International Coffee Organization [16], Brazil is the world's largest producer of coffee and this is an

* Corresponding author.
E-mail address: rkrohling@inf.ufes.br (R.A. Krohling).

https://doi.org/10.1016/j.inpa.2021.01.004

important commodity for the country. Biotic stress such as leaf miner, rust, brown leaf spot, and cercospora leaf spot, affect coffee plantations leading to the defoliation and reduction of photosynthesis, hence reducing the yield and quality of the final product [18].

Several efforts have been made using Artificial Intelligence (AI) to assist farmers to correctly identify the diseases and pests that affect their production and the severity of the symptoms. Computer-aided diagnosis (CAD) systems allow any farmer with access to a smartphone to enjoy expert knowledge in a practical and low-cost way [2]. Johannes et al. [6] developed a system capable of identifying diseases in photos of wheat leaves obtained by smartphones. Picon et al. [12] expands on this work using an adapted Deep Residual Neural Network-based algorithm to deal with the detection of multiple plant diseases from in-field acquired images.

Smartphone applications have been developed using deep learning methods. Elhassouny and Smarandache [3] developed a smart mobile application using convolutional neural networks (CNN) to recognize tomato leaf diseases. Toseef and Khan [17] developed a mobile app for diagnosis of crop diseases using a fuzzy inference system. Manso et al. [9] developed a system for quantification and classifying of leaf miner and coffee leaf rust. The system uses a threshold-based method to quantify the severity and to identify each symptom individually. Marcos et al. [10] and Boa Sorte et al. [14] used deep leaning approach for coffee leaf disease detection, both of them using CNN based architectures. Only a few diseases were considered by these authors, Marcos et al. [10] only dealt with rust and Boa Sorte et al. [14] with rust and cercospora leaf spot. In the mean time, Esgario et al. [4] proposed an approach based on deep neural network using single-task and multi-task system, capable of diagnosing four types of biotic stress and estimate its severity. Results for symptoms classification were greater than 97%.

In a recent study, Barbedo [1] explored the use of individual lesions instead of considering the entire leaf. The lesions were manually segmented, this allowed to significantly increase the dataset and to identify multiple and different lesions that affect the same leaf. However, manual segmentation is a laborious process specially in leaves with many small lesions. In this work our main contribution are twofold: (1) We expand our previous work on classification of diseases and symptoms of coffee leaves [4] aiming to develop a framework capable of automatically integrate image segmentation, cropping and classification process; (2) We develop and implement a smartphone application easy to be accessed and used by coffee farmers and experts.

The remainder of this paper is organized as follows. Section 2 presents the proposed method for segmentation and classification of coffee leaf symptoms and introduce the app development. Section 3 presents the experimental results. Section 4 draws some conclusions.

2. Materials and methods

The success of CNN for image classification and object detection problems motivated researchers to explore the capacity of these networks in solving semantic segmentation problems [5]. Since then, several architectures have been proposed in the literature with promising results in different areas of knowledge. In this work, two common semantic segmentation architectures in the literature were used, UNet [13] and PSPNet [20]. Several CNN architectures have been proposed, each with its particular characteristics. Despite the differences, they all share the same goal that is to increase accuracy and reduce the model complexity. Some architectures provide great performances on a wide range of applications. Some of the most common architectures used in the classification problem of plant diseases are: AlexNet, GoogLenet, VGG16, ResNet50, MobileNetv2.

Next, we present an overview of the proposed architecture, which is composed of two stages: a semantic segmentation stage trained on the segmentation dataset and the classification stage, trained on the symptoms dataset. The symptoms segmented from the first model were independently processed and classified in the second model. Thus, it is possible to identify multiple symptoms and return the exact severity of each one.

2.1. *Semantic segmentation architectures*

In this work we used two common semantic architetures in the literature: UNet and PSPNet. The UNet was proposed by Ronneberger et al. [13] initially for segmentation of biomedical images. The architecture consists of a fully convolutional network and can be divided into two main stages: contracting path and an expansive path. The contracting path is responsible for downsampling and follows a conventional neural network architecture, without the fully connected layer, i.e., an input image has its dimensionality and depth increased through convolution, ReLU, and max pooling operations. After the first path, there is an expansive path, which does the reverse process. The expansive path consists of upsampling operations and feature map convolutions, which are concatenated with corresponding feature maps from the contracting path.

The second architecture used was the Pyramid Scene Parsing Networks (PSPNet) proposed by Zhao et al. [20], which is a semantic segmentation network developed for the segmentation of complex scenes, in which global context information is important to distinguish similar objects. For example, a network that classifies a boat as a car, if this network could analyze the contextual information of the scene, it could perceive that the boat is over a river, which reduces the chances of a wrong classification. Initially, the image goes through a ResNet with the dilated convolution technique [19] to extract the feature maps. Then, the *pyramid pooling* module is applied to collect different representations of sub-regions, i.e., the activation map is partitioned into different sub-regions, and for each region the *average pooling operation* is applied. The outputs at the different levels of the *pyramid pooling* module contain feature maps of varying sizes, carrying contextual information at different scales. Then, the outputs go through the *upsampling* operation, and are concatenated with the original activation map produced by ResNet. Finally, they go through a last convolution step producing the final pixel-by-pixel prediction.

2.2. Dataset for semantic segmentation and symptom classification

The construction of a well-representative and well-labeled dataset is a key point in the performance of the model. This phase consists of three main steps, which are:

- Data collection: Different problems have different image acquisition difficulties. However, collecting images of plant diseases depends on external factors, such as time of year and location. In addition, the construction of any of these datasets requires a lot of manual effort to register the images and save them.

- Labeling: The labeling process consists of annotating the collected images, usually this process is assisted or carried out by a specialist.

- *Data augmentation* and pre-processing: Deep learning models need a large volume of labeled data to achieve good generalization. For this purpose, data augmentation techniques are used to expand the dataset and include more variations to the data, for example, through geometric transformations (rotation, mirroring, clipping, etc.) and intensity transformations (contrast, brightness and saturation). The preprocessing of the data consists of normalizing the images inputting the network. The most common techniques are resizing and subtracting colors by their average.

In a previous work, we construct a dataset named BRACOL [8], which is used in the current work.

2.3. Training

The next phase consists of training the model with the available dataset. The backpropagation algorithm used minimizes the error function by adjusting the network weights with the descending gradient method.

- Pre-training: It consists of initially training the CNN with some large database, for example, ImageNet (1.2 million images). The technique that reuses pre-trained models in one task in another is called *transfer learning*. The idea rests on the intuition that a model trained on a sufficiently large dataset is able to learn to extract visual characteristics that are useful for most problems.

- Fine tuning: The pre-trained model is then reused as a starting point for training the disease classification system. The last layer of the model is replaced by a layer compatible with the number of classes of the problem. The model in question is retrained with the image dataset using the backpropagation algorithm until it fits the data.

2.4. Segmentation and Symptom Classification

Extending the previous work by Esgario et al. [4], we have developed a framework for integrating the segmentation and classification tasks. In the segmentation stage, the model can isolate each symptom from the leaf and feed to a second network to classify each one. Fig. 1 shows an overview of the stages of the Symptom Segmentation and Classification (SSC) approach.

The first step is to acquire the image of the leaf that is resized and normalized before being inserted into the segmentation network. The semantic segmentation stage is formed by a CNN capable of isolating the background, the leaf and the symptoms, assigning a different label to each pixel of the image belonging to one of these classes. The symptom mask is partitioned, identifying each related component individually, i.e., each symptom. A bounding box is placed over each connected component of the symptom mask and the coordinates of these boxes are used to extract spots from the original image. These spots form a batch of images that are sent to the classifier. The classifier consists of a CNN trained to recognize the main symptoms of coffee diseases and pests. After the classification of each symptom image, it is possible, using the masks generated by the segmentation network, to calculate the severity of each symptom individually.

Since the segmentation and classification stages are independent, the training and validation of both were done separately and only then combined to form a whole system.

2.5. App Development and Implementation

The app's purpose is to integrate both segmentation and symptom classification methodology aiming to make it capable of identifying and estimating the stress severity caused by biotic agents on coffee leaves. As a result, any farmer with a smartphone and internet connection can receive a feedback on symptom severity and which diseases or pests are affecting the coffee leaves.

Embedding a CNN in a smartphone presents two main requirements: (1) the size of the network's weights that can be too large and may not fit on the device's memory; and (2) the need of computational resource to perform the model. Since not all smartphones can fulfill these requirements, we decided to deploy the CNN model on a server. Fig. 2 shows a schematic diagram of the developed system.

On the client side, we have a mobile application developed using *React Native*[1] framework, and *Expo SDK*.[2] The application sends coffee leaf images to the server. The server performs the CNN model and returns the diagnosis prediction. Finally the app displays it on the screen (Figs. 3 and 4).

The first back-end layer is based on the java web server *Tomcat* that implements a *Rest API* to be bought by the user as a service. All user information, for log purpose, is stored in a *MySQL* database. The second layer is based on *Flask*,[3] a framework based on *Python* that is designed for micro applications. It makes a direct execution of the machine learning models, which were developed also in *Python*. Every request for processing a new image that arrives at *Flask* is queued in *Redis*,[4] a *NoSQL* key-based database. If no data is being

[1] https://facebook.github.io/react-native/docs/getting-started.

[2] https://docs.expo.io/versions/latest/.

[3] https://flask-doc.readthedocs.io/en/latest/.

[4] https://redis.io/.

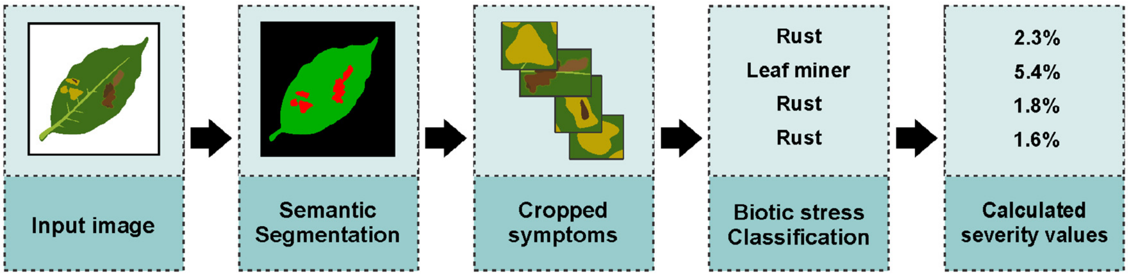


Fig. 1 – Main steps of the segmentation and classification of symptoms approach.

processed then the first available data of the *Redis* queue is read and sent to a previous trained model. The result of the model is then stored on *Redis* that will be further consulted by the user.

3. Results and discussion

The first part of this section consists of a detailed description of all databases created during the development of this work. The experimental results of semantic segmentation and classification using the coffee leaf image databases are presented. The experiments were conducted to assess how appropriate the CNN models were for the problems of classifying biotic stress and quantifying the severity. Details of the experimental configuration and the results obtained are presented in the following subsections.

3.1. *Experiments*

The database developed for this work contains images of Arabica coffee leaves affected by the main pests and diseases that affect coffee crops. A total of 1747 images of coffee leaves were collected, including healthy leaves and diseased leaves, affected by one or more types of pests and pathogens in different stages. The process of recognizing biotic stresses for labeling images was monitored by a specialist [4]. The database is composed of the four most common types of stresses that affect this crop, which are: leaf miner, rust, brown leaf spot and cercospora leaf spot.

From the obtained photos, two different databases were generated: (1) database of segmentation masks annotated at pixel level for three classes: background, leaf and symptom and (2) database with spots of each lesion. Details of each of the databases are described in the following.

3.1.1. *Segmentation dataset*

The construction of the segmentation database consists of a slow and demanding process of annotating the masks. A total of 500 images from the dataset of leaves were selected and the pixels of these images were annotated manually with one of three possible labels: background, leaf and symptom. These masks were considered as the ground truth and used in the training and evaluation of the segmentation model. Fig. 5 shows an example of a ground truth generated from a photo of a diseased leaf. Table 1 shows the division of samples by class of the segmentation database. From the 500 selected images, 100consists of images of healthy leaves and 400 of diseased leaves.

3.1.2. *Symptoms dataset*

This dataset was created from cropping of the apparent symptoms in the original images, to ensure that only a single stress was present in each image. A total of 2147symptom images were cropped. In addition to our images, 575 images made available by Barbedo [1] in his work were added to the database, totaling 2722 symptom images. Fig. 6 shows some examples of symptoms extracted from the original images. Table 2 specifies the number of images for each symptom in the dataset.

3.2. *CNN settings*

In order to meet the requirements of the CNNs, the input images were resized according to the database used. For the symptom dataset, images with a width and height of 224 pixels and 3 color channels were standardized, which matches the dimensions of the pre-trained networks on the ImageNet database. The segmentation database, on the other hand, were scaled to a width of 512 pixels, height of 256 pixels and 3 color channels. Since the segmentation networks used are invariant to the input size, the size $512 \times 256 \times 3$ was chosen arbitrarily, seeking to balance computational cost and quality of mask definition.

For the performance of all experiments, the proportions of 70%/15%/15% were used for the training, validation, and test, respectively. Data augmentation techniques were applied during the training, in such a way that with each new batch of images, new images are produced and propagated by the CNN. The weights of the network are adjusted until the most relevant discriminative features of that set of images are learned. To make training more efficient, the transfer learning technique was applied. The training was performed by fine-tuning pre-trained networks with the ImageNet database. The models were retrained end to end. The hyper-parameters used in all experiments in training the models are listed in the Table 3.

During the training of the networks, the states (weight set) were saved for the models with the lowest loss value for the validation set. The saved models were then evaluated using the test dataset and the results were calculated using common metrics in the literature. All experiments were programmed using PyTorch, a specific open-source library for

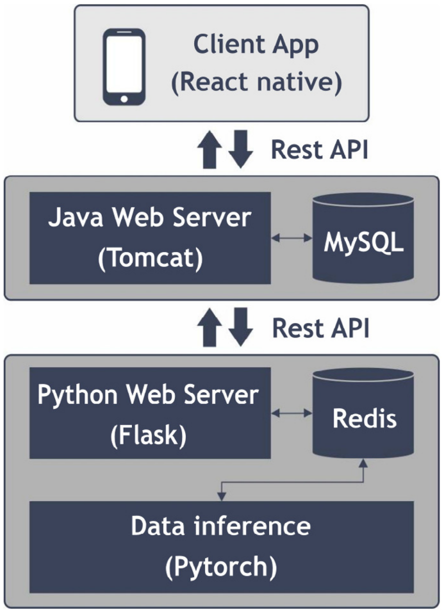


Fig. 2 – Schematic diagram of the smartphone app for coffee leaf diseases and pests diagnosis.

machine learning [11], and conducted on an NVIDIA GeForce GTX 1060 GPU with CUDA 10.0.

3.3. *Used metrics*

For the classification task, the following metrics were used: accuracy (AC), precision (PR), and recall (RE). Since a model capable of classifying more than two classes is being evaluated, the metrics precision and recall are calculated for each class individually and then its average is calculated, i.e., the classes are treated equally in such a way that they have the same weight in the final values.

The most common metrics used in the semantic segmentation task are: accuracy (AC) and mean intersection over union (abbreviated by MIoU) [5]. The calculation of the MIoU metric is described by the following equation:

$$MIoU = \frac{1}{k}\sum_{i=1}^{k}\frac{p_{ii}}{\sum_{j=1}^{k}p_{ij} + \sum_{j=1}^{k}p_{ji} - p_{ii}} \quad (1)$$

where p_{ij} is the amount of samples belonging to the class i predicted to be of the class j, i.e., p_{ij} is relative to the values of the confusion matrix generated from the results of classification, and k is equal to the number of classes of the problem. The main difference between semantic segmentation and classification metrics is that p_{ij} is now treated as pixels and not as samples.

In addition to the metrics presented, it is possible, using segmentation masks, to calculate the severity of diseases. The severity metric measures the percentage of the injured leaf area and can be calculated using the following equation:

$$Severity = \frac{A_{lesion}}{A_{leaf}} \times 100 \quad (2)$$

3.4. *Semantic segmentation*

In Table 4 the results obtained with the MIoU and AC metrics for the UNet and PSPNet architectures using the 50 test images of the segmentation dataset are presented. Among the architectures evaluated, UNet performed better than

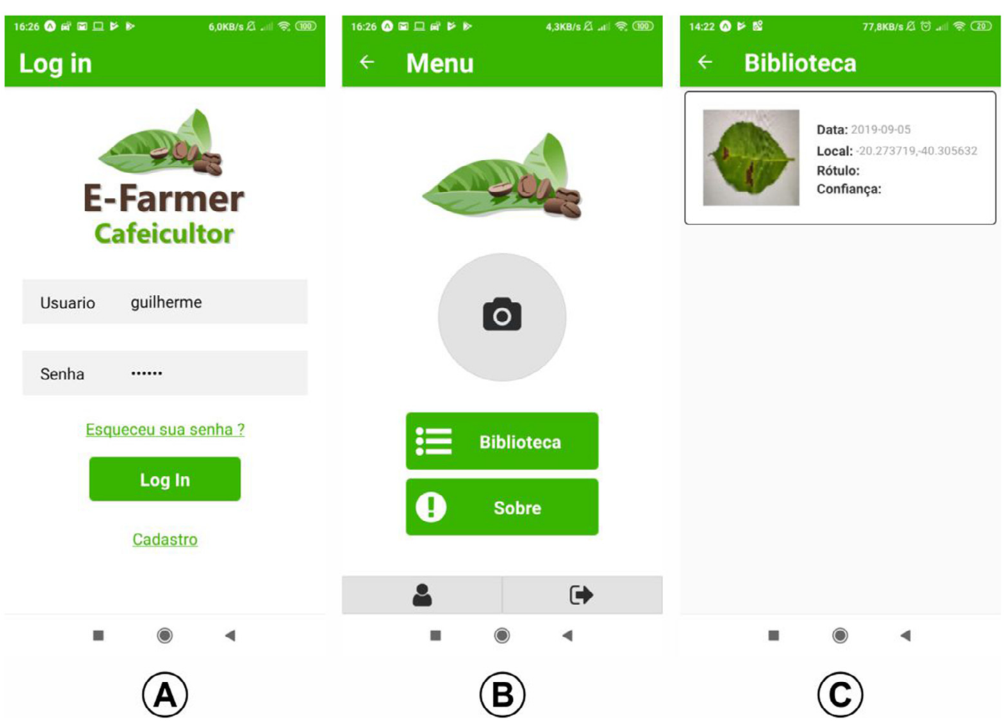


Fig. 3 – Screenshots taken while using the app: login (A); menu (B); and library (C).

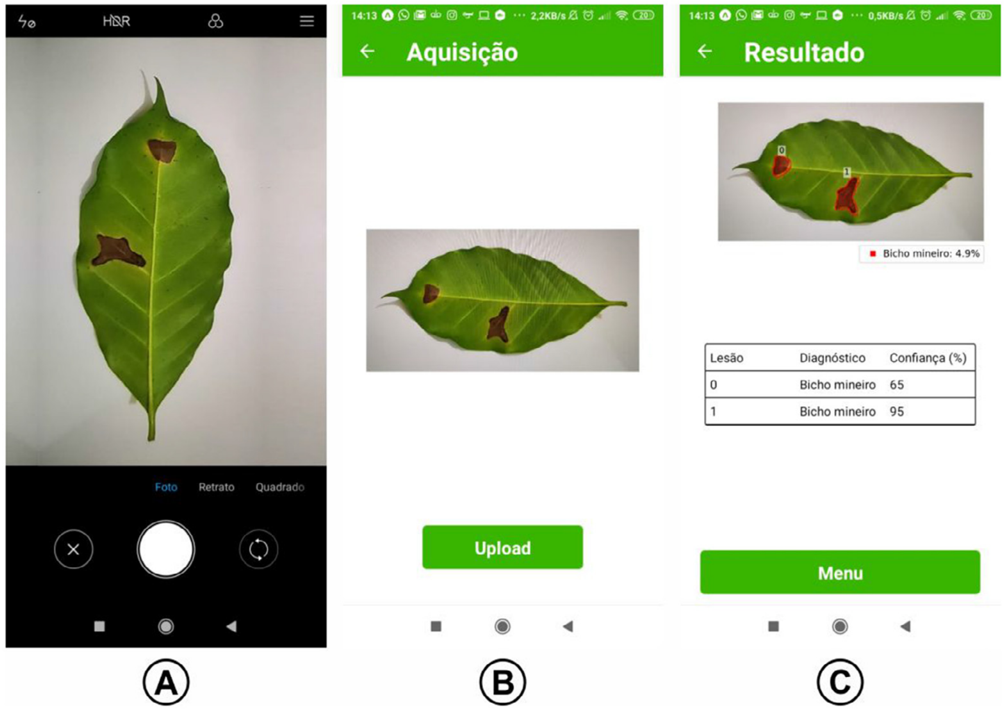


Fig. 4 – Screenshots taken while using the app: camera (A); verification and upload (B); and results (C).

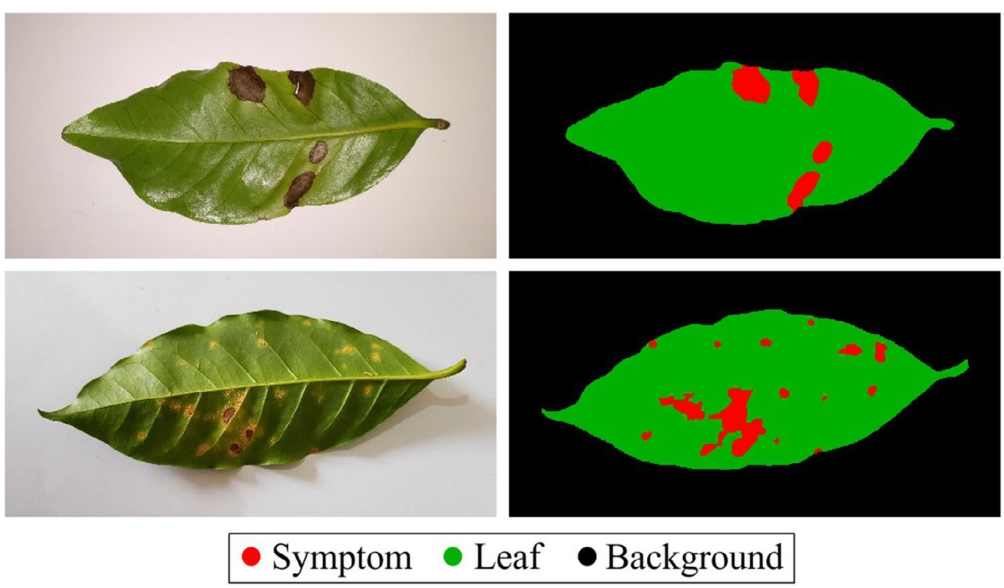


Fig. 5 – Examples of segmentation masks.

PSPNet, especially if compared by the MIoU metric, with a difference greater than 1%. Both results were good considering that even the ground truth showed inconsistencies in the demarcation of symptoms from the manual mask annotation process. These inconsistencies arose due to the subjectivity that exists when defining the limits between symptom and leaf. In addition, there was also an imprecision in the handling of the mask annotation tools used.

Since the segmentation masks provided an accurate calculation of the severity of symptoms, the severity for each sample was calculated from the ground truth masks (actual severity) and the masks generated by the CNN networks (predicted severity). From the severity values, the scatter plots were generated and the determination coefficients R^2 were calculated, in order to verify how close the data are to the regression line. The graphs are shown in Fig. 7.

Table 1 – Datasets details.

Biotic Stress	# Figures
Healthy	100
Leaf miner	117
Rust	119
Brown leaf spot	121
Cercospora leaf spot	33
Total	500

Table 2 – Dataset details.

Biotic Stress	# Figures
Healthy	256
Leaf miner	593
Rust	991
Brown leaf spot	504
Cercospora leaf spot	378
Total	2722

Both architectures presented a high correlation between predicted severity and true severity, with values of R^2 equal to 0.9798 and 0.9855 for the UNet and PSPNet networks, respectively. It is worth mentioning that, although PSPNet showed lower results for MIoU and AC, the severity values were more similar to those obtained from the ground truth masks. To verify the behavior of the networks in the generation of the masks, some results were selected from the test database to cover the worst and the best segmentation case. The segmentation results obtained for each of the selected images are shown in Fig. 8.

Visually analyzing the segmentation results obtained, it was possible to notice that the PSPNet presents softer and rounded masks when compared to the UNet. This tendency to smooth the masks leads to lower MIoU values when compared to those obtained by UNet. The images whose networks presented greater segmentation difficulties were those of leaves infected with rust, which is an expected behavior since the symptoms of rust sometimes present shades of yellow very similar to the color of the leaf tissue.

Finally, for each segmentation architecture, the average training times per epoch in seconds and the average inference times per image in milliseconds were measured. According to the results presented in Table 5, we can see that UNet is considerably faster during the training stage when compared to PSPNet. In contrast, the architectures did not show significant differences in inference time. In addition, the number of UNet parameters is much higher.

Table 3 – CNN training hyper-parameters.

Parameter	Value
Optimizer	Stochastic Gradient Descent
Loss function	Cross entropy
Epochs	100
Batch size	4 (512 × 256 × 3) and 32 (224 × 224 × 3)
Learning rate*	0.01
Momentum	0.9
Weight decay	0.0005

Decreases by a factor of 1/2 or 1/5 every 20 epochs, alternately.

Table 4 – Test results obtained with the UNet and PSPNet models for the segmentation database.

Model	MIoU (%)	AC (%)
UNet	**94.85**	**99.53**
PSPNet	93.69	99.31

3.5. *Symptoms classification*

For the symptom database, the following architectures were used: AlexNet, GoogLeNet, VGG16 and ResNet50 [4]. The best result was achieved with ResNet50 with accuracy of 97.07%, precision of 96.85% and recall of 96.69%. Fig. 9 presents the confusion matrix with the prediction results obtained by ResNet50.

Fig. 6 – Examples of symptoms images.

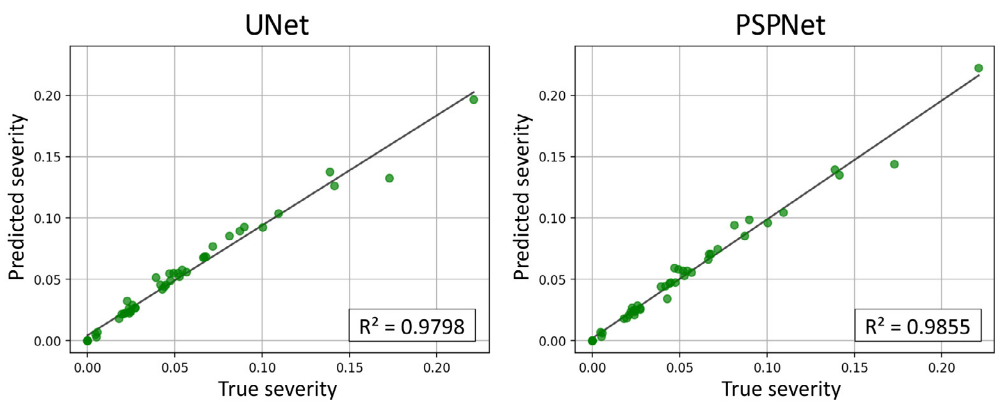


Fig. 7 – Scatter plot of actual versus predicted severity.

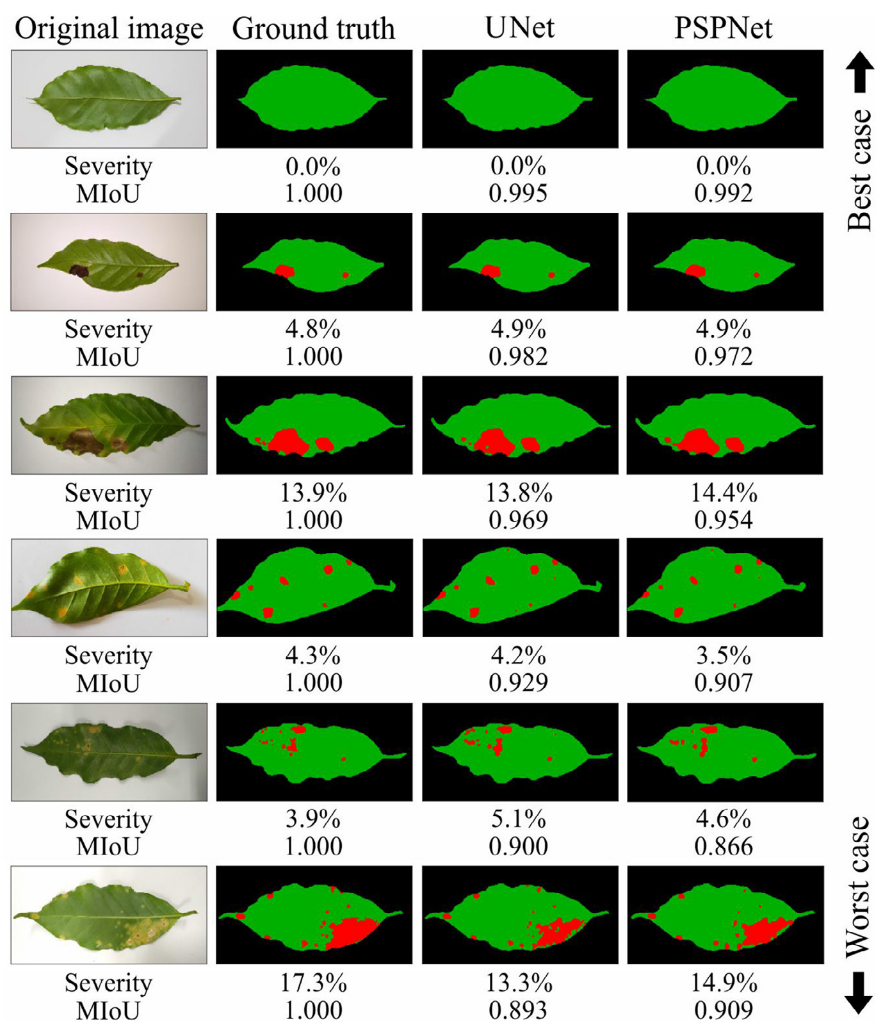


Fig. 8 – Visual segmentation results of some images of the test database.

Table 5 – Number of parameters and average training and inference time of the segmentation models.

Model	Parameters (M)	Training (s/epoch)	Inference (ms)
UNet	148.1	206	7.30
PSPNet	53.6	422	8.02

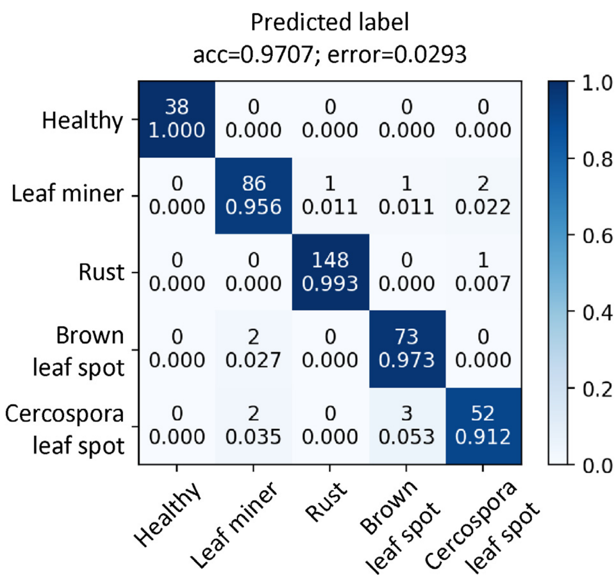


Fig. 9 – Confusion matrix with the prediction results using the ResNet50 architecture for the symptom database.

4. Conclusion

The methodology presented consisted of evaluating different approaches based on deep learning for the problem of segmentation, classification, and quantification of biotic stress of coffee leaves. The results of semantic segmentation reached a value of 94.85% for the metric mean intersection over union with the UNet. This result confirms that the model can reproduce masks very accurately compared to those generated manually. The results of the classification of biotic stress obtained with ResNet50 using images of symptom spots had an accuracy of 97.07%. The proposed approaches using convolutional neural networks obtained consistent results. Increasing the number of samples in the databases and adding new types of biotic stresses would increase the robustness and reliability of the system. Although this case study was focused on coffee leaves, the entire methodology is scalable to any other crops by using a new set of data images and adapting the network outputs to the number of classes desired. The development of this app provides a tool for growers for diagnosis of pests and diseases in coffee leaves using their smartphones. A limitation, however, of the framework would be the need for internet access for its operation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

J.G.M. Esgario and R.A. Krohling thank the funding support provided by Google Latin America Research Award (LARA 2018–2019). R.A. Krohling is also funded by the Brazilian research agency Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - grant n. 309729/2018-1. This work would not be possible without the support of J.F. Krohling who provided the opportunity to collect the coffee leaves during the year 2018 and part of 2019. We are also very thankful to H. Knidel who helped in the conception of this project and J.A. Ventura who provides us with valuable information about coffee diseases and pests.

REFERENCES

[1] Barbedo JGA. Plant disease identification from individual lesions and spots using deep learning. Biosyst Eng 2019;180:96–107.

[2] Dehnen-Schmutz K, Foster GL, Owen L, Persello S. Exploring the role of smartphone technology for citizen science in agriculture. Agron Sustain Develop 2016;36(2):25.

[3] Elhassouny A, Smarandache F. Smart mobile application to recognize tomato leaf diseases using convolutional neural networks. In: 2019 International Conference of Computer Science and Renewable Energies (ICCSRE). p. 1–4.

[4] Esgario JG, Krohling RA, Ventura JA. Deep learning for classification and severity estimation of coffee leaf biotic stress. Comput Electron Agric 2020;169:105162.

[5] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Garcia-Rodriguez J. A review on deep learning techniques applied to semantic segmentation. arXiv e-prints 2017; arXiv:1704.06857.

[6] Johannes A, Picon A, Alvarez-Gila A, Echazarra J, Rodriguez-Vaamonde S, Navajas AD, et al. Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput Electron Agric 2017;138:200–9.

[7] Kranz J. Measuring plant disease. In: Experimental techniques in plant disease epidemiology. Berlin, Heidelberg: Springer; 1988. p. 35–50.

[8] Krohling R, Esgario J, Ventura J. BRACOL - a brazilian arabica coffee leaf images dataset to identification and quantification of coffee diseases and pests; 2019. doi: 10.17632/YY2K5Y8MXG.1. https://data.mendeley.com/datasets/yy2k5y8mxg/1.

[9] Manso GL, Knidel H, Krohling R, Ventura JA. A smartphone application to detection and classification of coffee leaf miner and coffee leaf rust. arXiv e-prints 2019. arXiv:1904.00742.

[10] Marcos AP, Silva Rodovalho NL, Backes AR. Coffee leaf rust detection using convolutional neural network. In: 2019 XV Workshop de Visão Computacional (WVC); 2019. p. 38–42.

[11] Paszke A, Gross S, Chintala S, Chanan G. Pytorch. Computer Software Vers 2017;03:1.

[12] Picon A, Alvarez-Gila A, Seitz M, Ortiz-Barredo A, Echazarra J, Johannes A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. Comput Electron Agric 2019;161:280–90.

[13] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Munich, Germany: Springer; 2015. p. 234–41.

[14] Sorte LXB, Ferraz CT, Fambrini F, dos Reis Goulart R, Saito JH. Coffee leaf disease recognition based on deep learning and texture attributes. Procedia Comput Sci 2019;159:135–44.

[15] Suzuki N, Rivero RM, Shulaev V, Blumwald E, Mittler R. Abiotic and biotic stress combinations. New Phytol 2014;203(1):32–43.

[16] International Coffee Organization. ICO Annual Review; 2019.

[17] Toseef M, Khan MJ. An intelligent mobile application for diagnosis of crop diseases in Pakistan using fuzzy inference system. Comput Electron Agric 2018;153:1–11.
[18] Ventura JA, Costa H, Lima IdM. Manejo de pragas do café conilon. In: Ferrão RG, da Fonseca AFA, Ferrão MAG, de Muner LH, editors. Café Conilon. Vitória, ES, 2nd ed., Incaper; 2017. p. 434–79.
[19] Yu F, Koltun V, Funkhouser T. Dilated residual networks. In: Conference on Computer Vision and Pattern Recognition (CVPR). Havaí, EUA: IEEE; 2017. p. 472–80.
[20] Zhao H, Shi J, Qi X, Wang X, Jia J. Pyramid scene parsing network. In: Conference on Computer Vision and Pattern Recognition (CVPR). Hawaii, USA: IEEE; 2017. p. 2881–90.