Coffee Leaf Rust Detection Using Convolutional Neural Network

Alexandre Pereira Marcos
School of Computer Science
Federal University of Uberlândia
Email: alexandrepm2810@gmail.com

Natan Luis Silva Rodovalho
School of Computer Science
Federal University of Uberlândia
Email: natanlsr0@gmail.com

André R. Backes
School of Computer Science
Federal University of Uberlândia
Email: arbackes@yahoo.com.br

***Abstract*—Rust is a severe disease affecting many productive coffee regions. It is caused by a pathogenic fungi that attacks the underside of coffee leaves and it is characterized by the presence of yellow-orange and powdery points. If not treated, rust can cause a drop in coffee production of up to 45%. In this sense, this paper presents a contribution to the problem of rust identification that doesn't use "handcrafted" features, i.e., features extracted according to rules established by human programmers. Instead, we propose to train a Convolutional Neural Network (CNN) to learn to identify rust infection. We evaluated our CNN in a set of images provided by an expert and comparison results show that our approach is able to to detect the infection with a high precision, as corroborated by the high Dice coefficient obtained.**

***Index Terms*—texture analysis, convolutional neural network, coffee leaf rust**

I. Introduction

Nowadays, Brazil is the world's largest exporter of green coffee, accounting for 30% of world coffee production [1]. According to [2], pests and diseases are a very common problem in coffee farms. Among them, rust is a severe disease affecting coffee plantations in Brazil. Rust is caused by an endophytic fungus that attacks the leaf of adult coffee, especially older plants, and if not properly controlled can cause up to 45% reduction in coffee production [3].

In this context, image processing and machine learning techniques can help in the identification process of different plant diseases by speeding up the process or by detecting infections in early stages. In [4] the authors propose a spray robot to detect different leaf diseases using image processing in order to decide which pesticide to apply to the leaf. The work in [5] uses different machine learning techniques to detect rust infection in wheat leaves. A clustering approach can also be used to classify rust disease in wheat leaf images [6]. Rust detection is also the focus of the work proposed in [7], which uses graph pattern matching to detect signals of infection. The recognition and classification of visual symptoms of leaves affected by fungal disease is proposed in [8]. To accomplish that, the authors proposed a methodology based on Radon transform and SVM. In [9] the authors propose the use of a non imaging spectroradiometer in order to analyze the spectra of infected and non infected leaves.

In fact, computational methods can be very useful to evaluate characteristics present in the coffee leaves and, consequently, to facilitate and expedite the task of identifying coffee diseases, such as rust. Many computational techniques can be used in rust identification, such as different methods of image processing and texture analysis, besides machine learning techniques, such as deep learning [10], [11]. Deep learning is a set of machine learning algorithms that attempt to learn a pattern using different levels, each level corresponding to a different degree of abstraction. It has been used in several areas and for different types of data, such as images, sounds or texts. Deep learning techniques have demonstrated great advances in many works aiming pattern recognition in images, proving to be effective for both the generation and classification of patterns.

This paper aims to model and train a convolutional neural network for the detection of rust infection in coffee leaves. The remaining of the article is so organized as follows: Section II presents the concepts involved in a convolutional neural network and the Tensorflow library used to build the neural network. In Section III we describe the dataset used in the experiments. Section IV describes the experimental procedure used to detect rust, the network structure, how to perform data augmentation, and how we built training and validation sets. Section V reports the achieved results while conclusions and future work are presented in Section VI.

II. Convolutional Neural Network

Convolutional Neural Networks (CNN) are one of four categories of deep learning methods, along with restricted Boltzman machines (RBM), autoencoders, and sparse coding. The main characteristic in CNN is the existence of convolutional layers that act as receptive fields in the neurons, being its main application the processing of visual information [12]. Along with convolutional layers, CNNs also present pooling and fully connected layers [10], [13]. These three layers are

The authors gratefully acknowledges the financial support of CNPq (National Council for Scientific and Technological Development, Brazil) (Grant #301715/2018-1) and FAPEMIG (Foundation to the Support of Research in Minas Gerais) (Grant #APQ-03437-15). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brazil (CAPES) - Finance Code 001.

978-1-7281-5337-7/19/$31.00 ©2018 IEEE

what defined the main structure of any CNN and they are briefly described as follows.

The convolutional layer is responsible for applying a convolution operation to the input data. This operation acts as receptive field and emulates the response of an individual neuron to visual stimuli. The spatial extent of the receptive filter is specified by a filter size, usually tiling regions of size 3×3 or 5×5. Most CNNs apply an activation function after the convolutional layer. REctified Linear Unit (ReLU), $f(M) = \max(0, M)$ is preferable in comparison to other common activation functions (e.g., sigmoid and hyperbolic tangent) as it leads to much faster training in very deep architectures [14].

The pooling layer usually follows a convolution layer. This layer is responsible for reducing the size of the feature maps according to some criteria. A pooling of size 2×2 reduces the information of the region composed by 4 pixels to a single value by choosing either the maximum pixel (max-pooling) or the average pixel (average-pooling) of the region. Max-pooling is usually preferred as it presents faster convergence and better generalization for most cases [15].

The fully connected layer is placed as the last layers of the CNN. It is responsible for converting 2D features maps into a 1D feature vector, which will be the input of the last layer of the CNN, usually a softmax classifier. The softmax layer has N neurons, where N is the number of classes in the data set. The output of each neuron y_i is the probability of the input image to belong to class i [11]. The fully-connected layers contain about 90% of the total parameters (weights) of a CNN and are responsible for most of the computational cost of training.

A. Tensorflow

Tensorflow [1] is an open-source library used for machine learning applications and the creation of neural networks that mimic the human learning process. In computer vision, its use covers tasks ranging from classification to segmentation of images.

Developed by Google [16], the library includes several techniques of machine learning and deep learning. It has become popular for making easier the development of algorithms and the good feedback provided by the community. One differential of the library is that it allows its use in both CPUs and GPUs, the latter owning components specifically designed for matrix calculations. This makes the library more efficient for tasks that include image processing, or to process matrices containing different types of data. Moreover, the library is available for languages, such as Python and C++.

III. Dataset

In order to evaluate our CNN approach for rust detection we created a dataset composed by 159 images of coffee leaves. To acquire the images we used a Sony Cyber-shot DSC-W210 digital camera with 12.1 Megapixel sensor resolution, thus resulting in images with 2340×4160 pixels size. Each image contains a coffee leaf placed over a white paper, as shown in Figure 1(a).

After the acquisition each image was manually annotated by an expert. The expert was responsible to inform all spots containing the rust infection as a black and white mask, as shown in Figure 1(b). This step was necessary to assess the quality of the segmentation obtained by our CNN, as the idea is to train the CNN so that it must be capable to detect the same infection spots previously detected by the expert.

(a)

(b)

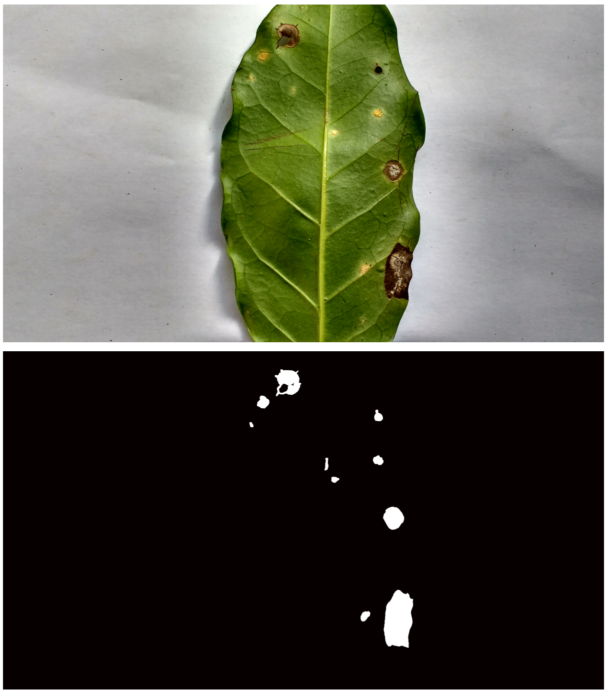

Fig. 1. (a) Sample of coffee leaf; (b) Regions indicating the presence of coffee leaf rust.

IV. Experiments

A. Network structure

Due to the small size of the images in our data set, we proposed a CNN structure with fewer layers than the traditional CNNs used in the literature. The Figure 2 shows the structure of our CNN. The network contains only two convolutional layers, which are followed by non-linearity ReLU filter and a max-pooling layer. To complement the network performance optimization, it was observed in the AlexNet [17] structure that a data normalization is applied after the max-pooling layer. Therefore, we chose to apply this normalization to each output of the max-pooling layers.

Given an input image of $40 \times 40 \times 3$ pixels size, the first convolutional layer filters the image with 32 filters with $5 \times 5 \times 3$ size kernels and applies ReLU non-linearity to the output. After max-pooling, the volume output size is $20 \times 20 \times 32$. At this volume we applied data normalization and a dropout of 80%, which consists of zeroing a part of the neurons, so that we prevent overfitting. The second convolutional layer filters

[1] http://tensorflow.org

the image with 64 filters with $5 \times 5 \times 3$ size kernels and applies ReLU non-linearity, max-pooling and data normalization, thus producing an output volume of size $10 \times 10 \times 64$. We use this resulting volume as input for the dense layers, the first using 2048 neurons and the activation function ReLU, and the second using 1024 neurons and the activation function SoftPlus. At this output we applied dropout of 20%. Finally, the output layer has two 2 neurons that determined the class (disease / normal).

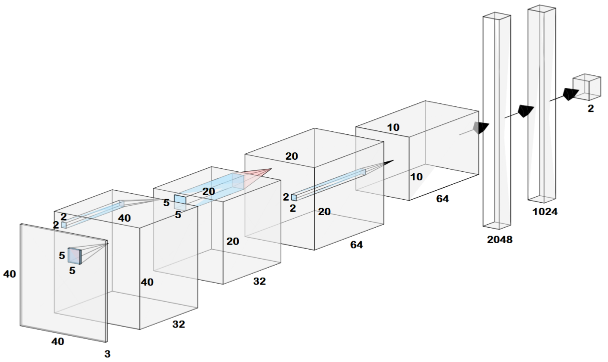

Fig. 2. An illustration of the architecture of our CNN.

B. Data Augmentation

To reduce overfitting of the network, we artificially enlarged the dataset by applying image transformations on the original images. Each texture image has 40×40 pixels size. For each image, we extracted 9 random patches of 37×37 pixels size. For half of these patches (selected at random), we altered the intensities up to 10% of the maximum intensity in the original image. Finally, we applied a scale transformation so that the patch is resized to the size of the original image, i.e., 40×40. We applied the same process to after performing a horizontal flip of the original image, totaling 18 variations for each sample original texture sample.

C. Training and validation sets

In order to train and validate our CNN we selected 4086 texture windows of 40×40 pixels size. We grouped the samples into two classes: disease, if the sample presents at least 50% its area compromised by coffee leaf rust; otherwise, the sample is classified as normal. Figure 3 shows some examples of texture windows used to train the network.

From 4086 samples we used 2859 to compose the training set, while the remaining 1227 samples compose the validation set. Due to the data augmentation process described in the previous section, the new training dataset contains a total of 51,462 images, i.e., 18 variations for each sample. In this work, we opted to use only the generated variations to train the CNN while the original images are used for validation. In this way, we can provide a large number of samples and variations of the patterns to be learned by the network, while the original dataset assures that the CNN is now capable of recognizing real data.

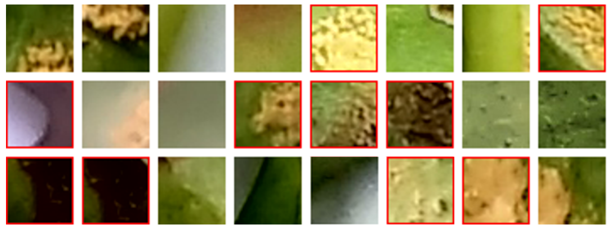

Fig. 3. Examples of texture windows used in the CNN. Samples of the area affected by the disease are shown with a red border.

V. Results

We performed the experiments on a PC with Intel(R) Core(TM) i5-4440 CPU @ 3.10GHz, 8GB RAM, 64-bit Operating System and NVIDIA GeForce GTX 1050 Ti with Tensorflow 1.10.0. The training procedure took 10003 global steps and 500 epochs and it achieved a 0.95 accuracy and 0.10 loss. After trained we used the CNN to classify windows of 40×40 pixels size of a coffee leaf image into two areas: infected (1) and non infected leaf (0). This resulted in a binary version of the image that we compared with the one provided by the expert. We used Dice coefficient D to measure how similar the two images are:

$$D = 2\frac{|A \cap B|}{|A| + |B|} \tag{1}$$

where A and B are two binary images and D, $0 \leq D \leq 1$, is how similar they are. The more the value D is close to 1, the more similar the images are.

Figure 4 show the Dice coefficient distribution yielded by our CNN approach (in blue). Results show that our approach is able to fully detect regions of the leaf affected by rust grossly, as we can see in Figure 5(a). We noticed that the CNN detects the regions surrounding the area affected by rust as also infected. Some may argument that this detection is correct as the detection of central region of infection is our main goal. Although this extra area detected is neighboring a region infected, its detection acts negatively in the Dice coefficient estimated. In this case, our CNN resulted in an average and median Dice coefficient of 0.59 and 0.61, respectively.

In the face of this evidence, we opted to evaluated the use of a post processing step to convert our coarse segmentation to a fine one. Thus, we applied a morphological erosion after the image segmentation using our CNN approach. For this, we used an erosion with a disk of radius $r = 13$ as structuring element. Although other radius values were evaluated, this radius was chosen as it presented the best resulting. The yellow area in Figure 4 shows the Dice coefficient distribution after applying the morphological erosion. By using this post processing step we are able to diminish the detected area, thus excluding the neighboring region while keeping only the central region of the infection, as shown in Figure 5(b). This results in a more precise detection of rust infection, which is confirmed by the increase of the Dice coefficient in the most of samples. By using the morphological erosion our CNN resulted in an average and median Dice coefficient of 0.79 and 0.82, respectively, a substantial improvement in comparison to the previous result.

VI. Conclusion and Future Work

In this paper we proposed a CNN-based methodology to segment coffee leaf rust, a severe disease affecting many productive coffee regions which is characterized by the presence of yellow-orange and powdery points. To accomplish that we modeled and trained a Convolutional Neural Network (CNN) to learn to identify signal of rust infection. We used this CNN to segment a set of 159 images provided by an expert. Results showed that our approach is able to detect the infected area of the leaf in a coarse way, and that a simple morphological erosion is able to improve its detection to a high precision, as corroborated by the high Dice coefficient obtained. As future work, we aim to explore other models of CNN (different types and the order of the layers) and their combination with other post processing techniques and how they impact on the detection of the rust infection and the computed Dice coefficient.

References

[1] "Produção mundial de café," 2016-2017. [Online]. Available: https://www.yarabrasil.com.br/nutricao-de-plantas/cafe/producao-mundial-de-cafe/

[2] Syngenta, "Dinâmica de pragas e doença do café," 2012.

[3] J. B. Matiello, *O café do cultivo ao consumo*, 1st ed. Editora Globo S. A., 1991.

[4] S. D. A, G. A. G, C. P. A, and K. P. L, "Intelligent autonomous farming robot with plant disease detection using image processing," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 1012–1016, 2016.

[5] M. Azadbakht, D. Ashourloo, H. Aghighi, S. Radiom, and A. Alimohammadi, "Wheat leaf rust detection at canopy scale under different lai levels using machine learning techniques," *Computers and Electronics in Agriculture*, vol. 156, pp. 119–128, 2019.

[6] D. Majumdar, A. Ghosh, D. K. Kole, A. Chakraborty, and D. D. Majumder, "Application of fuzzy c-means clustering method to classify wheat leaf images based on the presence of rust disease," in *FICTA (1)*, ser. Advances in Intelligent Systems and Computing, S. C. Satapathy, B. N. Biswal, S. K. Udgata, and J. K. Mandal, Eds., vol. 327. Springer, 2014, pp. 277–284.

[7] E. Lasso, T. T. Thamada, C. A. A. Meira, and J. C. Corrales, "Expert system for coffee rust detection based on supervised learning and graph pattern matching," *IJMSO*, vol. 12, no. 1, pp. 19–27, 2017.

[8] J. D. Pujari, R. Yakkundimath, and A. S. Byadgi, "Detection and classification of fungal disease with radon transform and support vector machine affected on cereals," *Int. J. of Computational Vision and Robotics*, vol. 4, pp. 261–280, 2014.

[9] D. Ashourloo, H. Aghighi, A. A. Matkan, M. R. Mobasheri, and A. M. Rad, "An investigation into machine learning regression techniques for the leaf rust disease detection using hyperspectral measurement," *IEEE J Sel. Topics in Appl. Earth Observ. and Remote Sensing*, vol. 9, no. 9, pp. 4344–4351, 2016.

[10] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[11] Z. Gao, L. W. 0001, L. Zhou, and J. Zhang, "Hep-2 cell image classification with deep convolutional neural networks," *IEEE J. Biomedical and Health Informatics*, vol. 21, no. 2, pp. 416–428, 2017.

[12] M. A. Ponti, L. S. F. Ribeiro, T. S. Nazaré, T. Bui, and J. Collomosse, "Everything you wanted to know about deep learning for computer vision but were afraid to ask," in *SIBGRAPI Tutorials*. IEEE Computer Society, 2017, pp. 17–41.

[13] Y. L. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[14] Y. LeCun, Y. Bengio, and G. E. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] D. Scherer, A. C. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks - ICANN 2010 - 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 6354. Springer, 2010, pp. 92–101.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," 2016.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105.

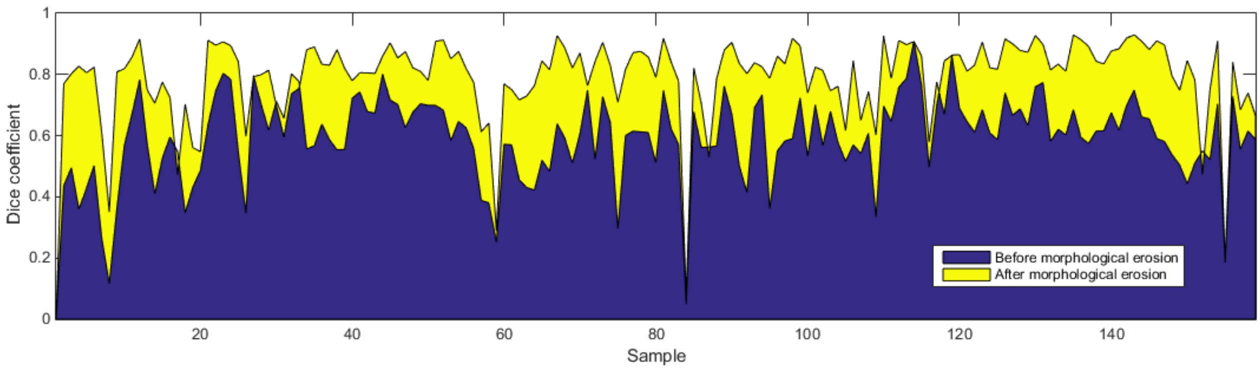

Fig. 4. Dice coefficient computed for each sample: Blue area: CNN result without morphological erosion; Yellow area: CNN result with morphological erosion.

(a) (b)

Fig. 5. Rust regions detected by the CNN: (a) Before morphological erosion ($D = 0.36$); (b) After morphological erosion ($D = 0.81$).