Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 135–144

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Coffee Leaf Disease Recognition Based on Deep Learning and Texture Attributes

Lucas Ximenes Boa Sorte[a], Carolina Toledo Ferraz[a],

Francisco Fambrini[b], Roseli dos Reis Goulart[c], José Hiroki Saito[a,b]

a-UNIFACCAMP, University Center of Campo LimpoPaulista-SP, Brazil
b-Federal University of São Carlos, São Carlos-SP, Brazil
c - Federal Institute of South Minas Gerais – Agriculture Department, Muzambinho-MG, Brazil

Abstract

An automatic coffee plant disease recognition system is required since coffee is an important commodity in the world economy and its productivity and quality are affected by diseases such as Cercospora and Rust. This research aims to apply computational methods to recognize main diseases in coffee leaves, with the purpose to implement an expert system to assist coffee producers in disease diagnosis during its initial stages. Since these two diseases are shapeless, it inspires a texture attribute extraction approach for pattern recognition. Two texture attributes were considered in this work: statistical attributes and local binary patterns. The texture attribute vector were computed for a collection of images of coffee leaves and used as input to a feedforward neural network. The results were compared with the recognition rate of a convolutional neural network with deep learning applied directly to the same collection of images, without extraction of texture attributes. Surprisingly, this second approach showed better results than the texture extraction method. It could be explained by the small number of diseases we aimed to recognize and a sufficient number of training samples used during the deep learning process. The best Kappa coefficient obtained was 0.970, and sensitivity was 0.980.

*Keywords:*Thermography; Image Processing; coffee leaf disease; cercospora; rust; Diagnosis

10.1016/j.procs.2019.09.168

1. Introduction

Coffee disease recognition, in its initial stage, is very important since losses could be avoided, contributing to the productivity and quality of this commodity plantation [1]. Coffee Leaf Rust disease, caused by the *Hemileia vastatrix* fungus, and *Cercospora*, known as Brown Eye Spot, caused by the *Cercospora coffeicola* fungus, are two terrible diseases that must be quickly detected. The Brown Eye Spot is shown in Figure 1(a) and the Coffee Leaf Rust is shown in Figure 1(b).

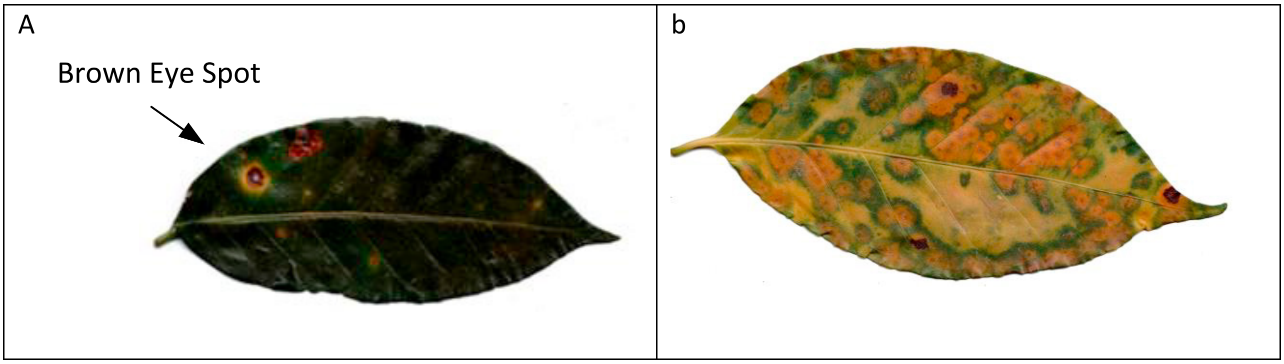


Figure 1 – Coffee leaf diseases: (a) Brown Eye Spot, caused by *Cercospora coffeicola,* and (b) Rust, caused by *Hemileia vastatrix*.

Brown Eye Spot causes brown rounded necrotic lesions, with dark centre, surrounded by a yellow halo. The affected leaves fall rapidly and the branches dry, causing a reduction in plant productivity and lowering the fruit's quality. This disease also causes intense leaf fall and poor plant growth [2].

Leaf Rust disease is characterized by shapeless lesions that cause early leaf falls, greatly affecting the year's productivity. On average, 35% of the whole coffee plantation is commonly affected when weather conditions are adequate for the disease, and in prolonged dry seasons, with high occurrence of fungus, the damage can exceed 50% [3].

To avoid these problems, an automatic coffee plant disease recognition system development is proposed, aiming to detect Cercospora and Rust diseases in coffee leaves. Since the diseases are shapeless, it inspires a texture attribute extraction procedure before the disease recognition step. Two texture attributes were considered in this work: statistical attributes and local binary patterns. On the other hand, since convolutional neural networks with deep learning have been recently applied to a wide range of problems [4], a direct application of this architecture to the coffee plant disease recognition task was also investigated. The main contribution of this research is the conduction of experiments using attribute vectors with relatively small number of input elements for the neural network to classify the coffee leave images, compared to experiments using the image pixels directly to the deep learning convolutional neural networks.

Some related works have been published previously, such as the following. Suhartono et al. presented a method using fuzzy logic-based expert systems, and decision tree using a hierarchical classification [1]. Gutte et al. described a method of plant diseases classification using K-means clustering technique to segment the affected part, and then using support vector machine for disease classification [5]. Mengistu et al. described a method of disease detection on coffee leaves based on neural networks and K nearest neighbour clustering [6]. Sladojevic et al. described the use of deep learning neural networks capable to recognize 13 different plant diseases, with the ability of distinguishing the attributes of leaf surroundings [7]. Fuentes et al. published an experiment considering three main families of detectors: faster region-based convolutional neural network (Faster R-CNN), region-based fully convolutional network (R-FCN), and single shot multibox detector (SSD), for the recognition of tomato diseases [8]. These and other interesting articles have different focus of this proposal.

This paper is divided in the following sections. In Section 2, we describe the texture attribute detection methods and deep learning convolutional neural network architecture, both used in this work. In Section 3, we present the coffee leaf image database and the evaluation coefficients. In Section 4, experiments and results are presented, followed by Section 5 of conclusions and future works.

2. Texture Attribute Detection and Deep Learning Architecture

In this section, we describe two texture attribute extraction approaches: the statistical attribute extraction and local binary pattern. Then, a convolutional neural network with deep learning used in this work is presented.

2.1 Statistical Texture Attribute

One important statistical attribute extraction approach is based on the Gray Level Co-Occurrence Matrix (GLCM) proposed by Haralick et al. [9]. GLCM is a matrix that considers the counting of the number of neighbouring pixels with a given gray level that occur considering a reference pixel with its own gray level. Each element of GLCM is given by a specific pair of gray level values, observed in the reference pixel and its neighbouring pixel, in the original image. Besides, each GLCM is computed considering a specific direction formed by these two pixels.

Figure 2(a) illustrates an original image, with some pairs of gray level values taken from the reference and neighbour pixels, considering the horizontal direction, left to right. Figure 2(b) refers to the corresponding GLCM that counts the number of pixel pairs, with the same pair of values of reference × neighbour pixels. Therefore, the numbers 2 and 3 showed in Figure 2(b) correspond to the number of occurrences of the each corresponding pair of gray level values in the original image.

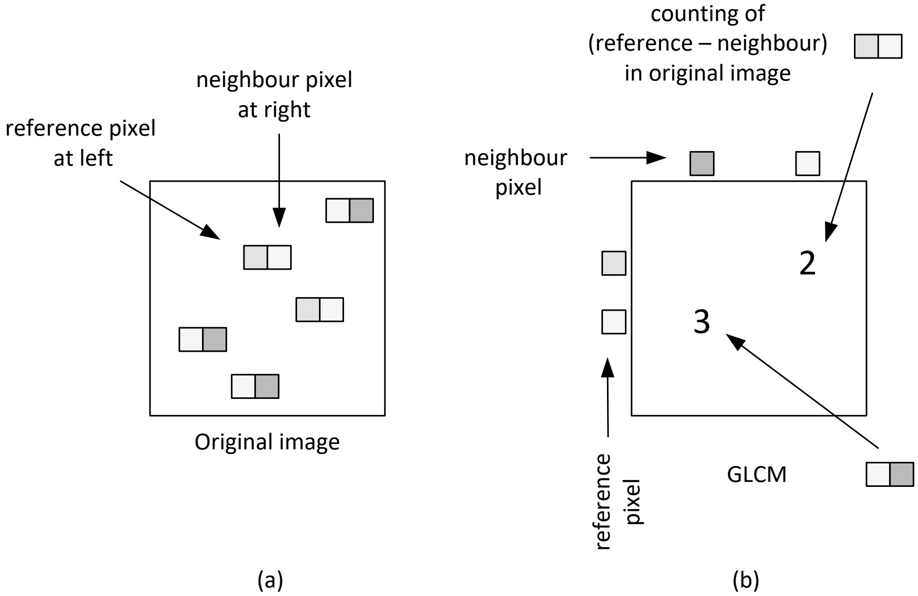


Figure 2.GLCM illustration: (a) original image, (b) GLCM.

Some statistical attributes are computed using GLCM, such as energy, correlation, contrast and homogeneity, that are used to characterize image textures. Equation (1) corresponds to the contrast computation, where *i* and *j* are the GLCM coordinates, and *p(i,j)* is the GLCM value after its normalization.

$$\sum_i \sum_j (i-j)^2 p(i,j) \qquad (1)$$

Equation (2) refers to correlation, where σ_i, σ_j, μ_i, and μ_j, are the standard deviation on line *i* and *j*, and average in *i* and *j,* respectively.

$$\frac{1}{(\sigma_i \sigma_j)} \sum_i \sum_j (i-\mu_i)(i-\mu_j) p(i,j) \qquad (2)$$

The following equation (3) refers to energy computation, and equation (4), to homogeneity.

$$\sum_i \sum_j p^2\,(i,j) \tag{3}$$

$$\sum_i \sum_j \frac{p(i,j)}{(1+|i-j|)} \tag{4}$$

2.2 Local Binary Pattern

Another texture attribute measurement can be obtained considering a small region, such as *3×3*, of one central pixel and its eight neighbours, proposed by He and Wang [10], and its binary version, Local Binary Pattern [11], computed using equation (5):

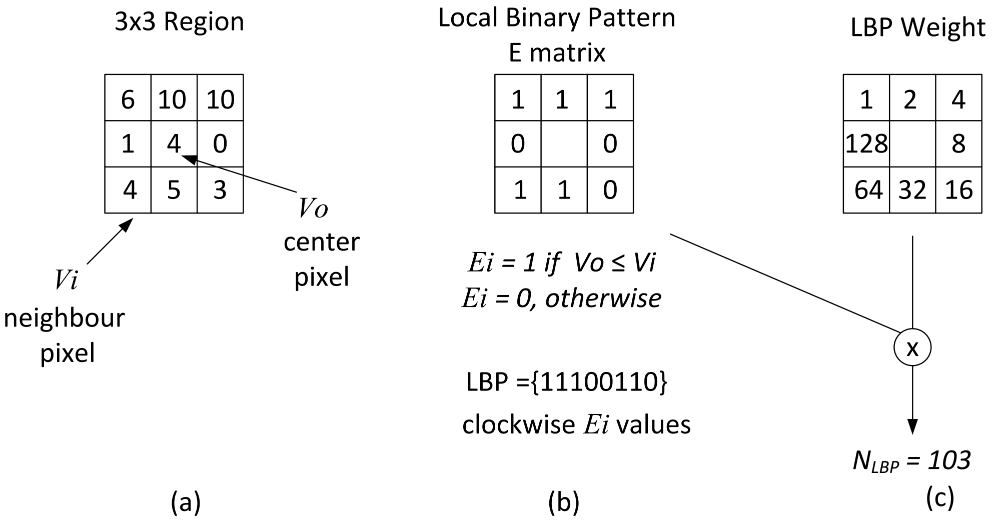
$$N_{LBP} = \sum\nolimits_{i=1}^{8} E_i . 2^{i-1} \tag{5}$$

where N_{LBP} is the LBP value of the region being considered. In LBP, the central pixel is called *Vo,* and the neighbouring pixels *Vi,* with $i = 1,\dots,8$, clockwise, with $i = 1$, top left position. For each neighbour position *i,* E_i is computed using equation (6):

$$E_i = \begin{cases} 1\ if\ V_0 \le V_i \\ 0, otherwise \end{cases} \qquad (6).$$

Each E_i is multiplied by its weight 2^{i-1}, and the weighted sum is computed to get N_{LBP}. The concatenation of the E_i values, with $i = 1,\dots,8$, is the so-called LBP, a local binary pattern.

In Figure 3(a) it is shown a *3×3* region and its pixel values. The central pixel has a value of 4. Figure 3(b) illustrates the E_i values computed for the *3×3* region, and the resulted LBP pattern is highlighted. In Figure 3(c), it is shown the respective weights and the weighted sum N_{LBP}, with value 103.



6	10	10
1	4	0
4	5	3

1	1	1
0		0
1	1	0

1	2	4
128		8
64	32	16

Figure 3. (a) 3x3 region of the image, (b) Local Binary Pattern computation, and (c) weighted sum N_{LBP}.

Ojala, Pietikäinen, and Mäenpää [12], proposed a rotation invariant version of LBP. In their proposal, texture attributes are computed considering P neighbouring pixels in a radius R from the central pixel, as described in equation (7):

$$LBP(P,R) = \sum_{p=0}^{P-1} s(q_p - q_c)2^p \qquad (7).$$

Considering q_c as threshold value, each neighbour binary value can be obtained by equation (8)

$$s(q_p - q_c) = \begin{cases} 1 \; if \; 0 \le (q_p - q_c) \\ 0, \; otherwise \end{cases} \tag{8}$$

where q_c corresponds to the central pixel and q_p, with $p = 0,..., P - 1$, corresponds to a neighbouring pixel, equidistant to the central pixel, resulting in a symmetric and circular neighbourhood. The coordinates of a given neighbouring pixel is given by ($x_c + R\ cos(2\pi p\ /\ P)$, $y_c - R\ sin(2\pi p\ /\ P)$), where ($x_c$, y_c) are the central pixel coordinate. An example of the rotation invariant LBP is shown in Figure 4(a), with neighbourhood $P = 8$ and radius $R=1.0$; Figure 4(b) shows an example with $P = 12$ and radius $R= 2.5$; and in Figure 4(c), $P = 16,$ radius $R = 4.0$. This second kind of LBP is rotation invariant since all neighbouring pixels is equidistant to the central pixel, and two LBP patterns only different by rotation are considered equivalent.

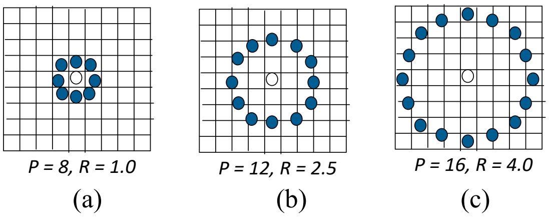


Figure 4. Examples of symmetric neighbourhood samples of rotation invariant LBP: (a) for 8 neighbours, (b) 12 neighbours, and (c) 16 neighbours, with radius 1.0, 2.5, and 4.0, respectively.

Since a LBP pattern is composed by a sequence of binary values such as the one shown in Figure 3(b), a simplified set of LBP patterns denoted as uniform is defined as a set of LBP patterns that have only one transition of the sequence of binary values, e.g. {00001111} and {1100000}. If the pattern has more than one transition is non-uniform, e.g. {00110011} and {01011010}. In rotation invariant LBP, a pattern {00001111} is equivalent to the sequences {00011110}, {00111100}, {01111000}, {11110000}, {11100001}, {11000011}, and {10000111}, since these patterns are only different by rotation.

2.3. Deep Learning AlexNet

Convolutional Neural Networks, also denoted ConvNets, have been used intensively, with its increased processing capacity, in high data volume processing. AlexNet architecture, proposed by Krizhevsky et al. [4], was considered by Szedegy [13] a hallmark for ConvNets. This architecture was submitted to the ImageNetILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) in 2012 and outperformed the second placed applicant, with an architecture used for the ImageNet database with more than a thousand possible classes. AlexNet architecture is based on LeNet [14], which is an evolution of Neocognitron, the first convolutional neural network proposed by Fukushima [15].

In the work of Vargas et al. [16], AlexNet showed better results for pedestrian detections than other techniques. In the work of Sousa et al. [17], the authors highlighted the AlexNet effectiveness in the guarana plant classification.

Figure 5 shows the configuration of AlexNet, composed by 8 layers, where the first 5 layers are convolution layers, and the last 3 layers are full connected to the previous layer outputs. The convolutional layers Conv 1, Conv 2 and Conv 5 are followed by the maxpooling of 3x3 filters to downsample the width and height of the tensors while keeping the same depth. A convolution refers to a filtering of a small region of the image in previous layer to detect the characteristic contained in that area, denoted receptive field. Maxpooling is an operation to reduce the characteristic matrix resulted by the convolution layer, selecting the greatest value of input in each considered region. The last layer, Softmax classifier, is full connected with 1000 outputs. The ReLU (Rectified Linear Unit) function is used as the activation function, which showed to be more efficient than the sigmoid and hyperbolic tangent activation functions.

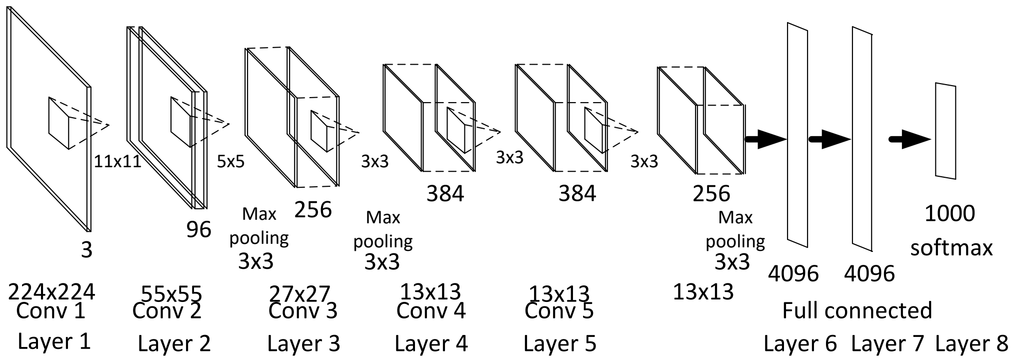


Figure 5. Simplified AlexNet architecture layers, with 5 convolutional layers and 3 fully connected layers.

3. Coffee Image Database and Evaluation Coefficients

In this section, we describe the Coffee Leaf Image Database, followed by the evaluation coefficients used in this paper for validation of the experiments.

3.1 Coffee Leaf Database

The images in the database were obtained from a coffee plantation farm belonging to the Federal Institute of South Minas Gerais – Agriculture Department, Muzambinho-MG, scanned with 300 dpi resolution. The injured parts were obtained as 128x128 pixels subimages. We obtained 750 subimages for each class, Cercospora, Rust, and Healthy leaves, using some data augmentation techniques such as mirroring, and random crops. The number of raw images, before augmentation, was about 500 images for each class.

In Figure 6, ten samples of each class of diseases to be recognized are shown, along with ten samples of healthy leaves. It can be observed that the affected part of the leaf is differently distributed, making the learning process difficult by an automatic pattern recognition system.

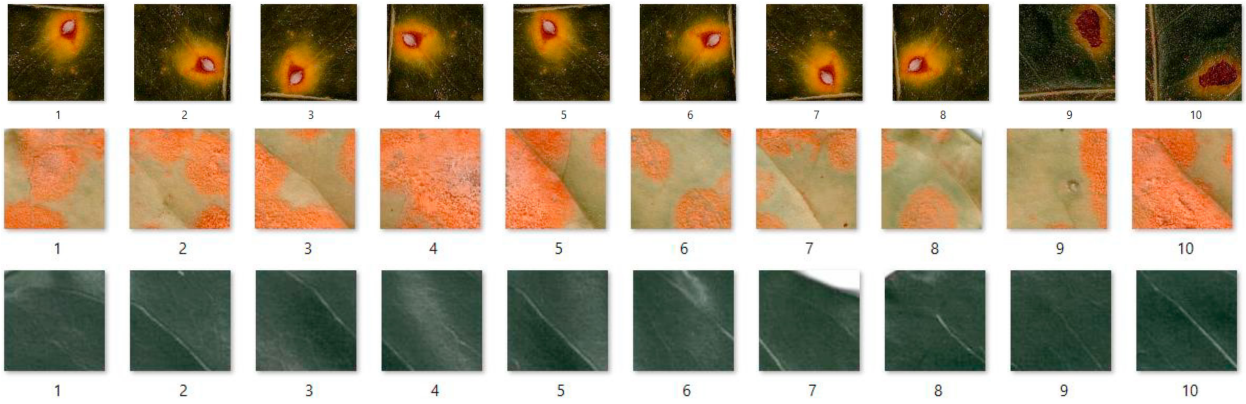


Figure 6. Examples of Coffee Leaf Database samples: (a) *Cercospora coffeicola*, (b) *Hemileia vastatrix* (rust) and (c) Healthy leaves.

3.2 Kappa and Sensitivity

Among several evaluation indexes, one that is commonly used is the Kappa index, *K*, proposed by Jacob Cohen [18]. Equation (9) described how K is computed:

$$K = \frac{p_o - p_e}{1 - p_e}, \tag{9}$$

where p_o and p_e are obtained using equations (10) and (11), respectively:

$$p_o = \sum_{i=1}^{r} \frac{n_{ii}}{n}, \tag{10}$$

$$p_e = \sum_{i=1}^{r} \frac{n_{i*}\ n_{*i}}{n^2} \tag{11}$$

In equation (10), n_{ii} is the number of matches by two evaluators. In the case of supervised learning using a neural network, one evaluator can be the neural network, and the other, the specialist, who defines a ground truth. Variable r is the number of classes and n is the number of input samples to be evaluated. In equation (11), the variable n_{i*} represents the number of evaluations by the first evaluator as class i; and n_{*i}, represents the number of evaluations by the second evaluator as class i. The interpretation of the Kappa coefficient is summarized in Table 1.

Table 1. Kappa coefficient interpretation.

Kappa coefficient Interval	Level of agreement	% of data that are reliable
[0, 0.2]	None	0 – 4%
[0.21, 0.39]	Minimal	4-14%
[0.40, 0.59]	Weak	15-35%
[0.60, 0.79]	Moderate	36-63%
[0.80, 0.90]	Strong	64-81%
[0.91, 1.00]	Almost perfect	82-100%

Another important coefficient, sensitivity or recall, is computed as True Positives, TP, divided by the sum of TP and False Negatives, FN, as shown in equation (12) [19]. Sensitivity is also known as true positive rate and revocation index.

$$Sensitivity = \frac{T_P}{T_P + F_N} \tag{12}$$

4. Experiments and Results

Seven experiments were conducted, first with texture attribute computing, followed by a neural network training and testing; and four experiments were conducted using the deep learning approach.

4.1 Texture Attribute Experiment

The experiments were first conducted on TBDR (Texture Based Disease Recognition) approach. The statistical attributes contrast, correlation, energy and homogeneity were considered. For the LBP attributes, we used the rotation invariant approach, as described in Section 2.2. Uniform and non-uniform LBP were also considered.

After texture attribute extraction, we applied a feedforward neural network to our data, known as *Patternnet* [20], for training and testing. It was used the default train function – trainscg (scaled conjugate gradient backpropagation); the default performance function, cross entropy; and the number of hidden neurons 10, and 15, at the presented experiments. The number of training samples was 250 in five first experiments, and 550 in two last experiments; and the number of testing samples was 50 for each class in all experiments. The number of 250 samples showed adequate for training. The two experiments with 550 samples were conducted to compare the results with the experiments using deep learning, that used this number of samples.

In Table 2, we present the seven experiments we conducted, highlighting the differences from one another. Since in *Patternnet* the number of hidden neurons can be varied, the seven experiments were executed several times varying the number of hidden neurons, from 8 to 20. The results obtained using 10 hidden neurons, and 15 hidden neurons are shown as follows.

Table 2 – Texture attribute (TBDR) experiments.

	Texture attribute vectors	Number of training Samples
Exp 1	Statistical attributes after histogram equalization	250
Exp 2	Rotation invariant uniform LBP	250
Exp 3	Rotation invariant non uniform LBP	250
Exp 4	Rotation invariant uniform LBP after histogram equalization	250
Exp 5	Statistical attributes and uniform LBP	250
Exp 6	Statistical attributes after histogram equalization	550
Exp 7	Rotation invariant uniform LBP	550

In Table 3, we can see the TBDR results considering 10 hidden neurons. The best value of both Kappa and sensitivity indexes was obtained in Experiment 3, of rotation invariant non uniform LBP texture attributes, with Kappa = 0.840 and sensitivity = 0.893. The low results of Experiments 6 and 7 that correspond to the 550 training samples can be explained by the occurrence of overfitting.

Table 3 – TBDR results of Experiments 1-7 with, 10 hidden neurons.

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
Kappa	0.720	0.810	**0.840**	0.800	0.810	0.180	0.600
Sensitivity	0.813	0.873	**0.893**	0.870	0.873	0.453	0.733

In Table 4, TBDR results with 15 hidden neurons are a little better for all seven experiments. The best value for both indexes was also obtained in Experiment 3, of rotation invariant non uniform LBP texture attributes, with kappa = 0.900, and sensitivity = 0.933. These results were also obtained in Experiment 5, which used an enhanced attribute vector that corresponding the LBP attributes of Experiment 3 along with the statistical attributes of Experiment 1.

Table 4 – TBDR results of Experiments 1-7 with, 15 hidden neurons.

	Exp1	Exp2	Exp3	Exp4	Exp5	Exp6	Exp7
Kappa	0.790	0.830	**0.900**	0.860	**0.900**	0.320	0.710
Sensitivity	0.860	0.886	**0.933**	0.906	**0.933**	0.546	0.806

This network used as input an attribute vector that in the case of statistical attributes has four elements, uniform LBP, ten elements, and 36 in non-uniform LBP. The results of TBDR should be compared with the following experiments with the deep learning approach applied directly to the image samples.

4.2 Deep Learning Experiment

In this subsection, we describe experiments using deep learning on the leaf image samples to recognize healthy and infected leaves. The Deep Learning Disease Recognition (DLDR) experiments were conducted using a modification of AlexNet, as summarized in Table 5. The number of training samples are specified in this table, and the number of test samples was 50, in all the DLDR experiments.

A modified AlexNet neural network, with changes that yielded performance gain with a small number of samples was used, with jpg compressed images, of 128x128 pixels, as input. The hidden layers were composed by three convolutional layers, with radius 2 (5x5) filters, and ReLU activation function. The first hidden layer had 42 filters, the second one, 74, and the third hidden layer, 74. All these layers had a MaxPooling of size 2. The fully-connected output layer had 812 neurons, using the SoftMax as the activation function and a dropout of 20%.

The network was trained with 20 epochs, using the backpropagation algorithm. As an optimizer, the Adam function was used with learning rate of 0.001. The validation process was performed on run time and applied to 10% of the input samples.

Table 5.DLDR experiments using deep learning.

	Number of training samples	Image type
Exp 8	550 samples of each class	Gray scale
Exp 9	550 samples of each class	RGB
Exp 10	550 samples for healthy leaf; and 700 samples for disease affected classes	Gray Scale
Exp 11	550 samples for healthy leaf; and 700 samples for disease affected classes	RGB

Experiments 8 – 11 were conducted, varying the number of training samples and type of images, gray scale and RGB. The results of Kappa and sensitivity index are shown in Table 6. We can note that the best values were obtained in Experiment 11, with Kappa = 0.970, and sensitivity = 0.980.

Table 6 Results of DLDR

	Exp8	Exp9	Exp 10	Exp 11
Kappa	0.661	0.661	0.895	**0.970**
Sensitivity	0.773	0.773	0.933	**0.980**

4.3 Analysis of the Results

Comparing the results of Tables 3 and 4 for TBDR, and Table 6 for DLDR, the best result can be attributed to DLDR approach since: 1) the best TBDR kappa coefficient is 0.900, versus 0.970 for DLDR, and (2) the best sensitivity value is 0.933 for TBDR, and 0.980, for DLDR. Another result is that increasing the number of training samples from 250 to 550, in TBDR, the evaluation indexes decreased drastically, probably because of overfitting. On the other hand, increasing the number of samples from 550 to 700, improved the DLDR performance, probably because the small number of three classes favoured the DLDR.

5. Conclusions and Future Works

This work aimed to investigate computational methods to recognize diseases in coffee leaf images, aiming to assist coffee producers to increase their productivity and quality of their commodity. We conducted experiments for recognizing three output classes: Cercospora, Rust, and healthy leaves. Two approaches were compared: 1) TBDR, using texture attribute vectors as input to a neural network classifier using statistical and local binary attributes, and 2) DLDR, using a convolutional neural network with deep learning applied directly to the sample images. The Kappa coefficient and sensitivity rates were computed for each experiment to provide information for comparisons. The best result using the TBDR approach was obtained using the local binary pattern texture attribute computation, with kappa = 0.900, which is considered as a strong agreement in Table 1, and a sensitivity of 0.933. Using DLDR, the best result for Kappa was 0.970, considered an almost perfect agreement in Table 1, and sensitivity was 0.980. These results showed that both methods are feasible in the development of a coffee leaf disease recognition system to assist coffee producers. As future works, other type of coffee diseases should be studied using these two approaches, including other part of coffee plants, such as stem and root. Also, other types of texture attributes, such as spectral attributes should be verified. On the other hand, other experiments with different image samples, should be conducted.

Acknowledgements

The authors gratefully acknowledge CAPES (Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior), Ministry of Education, Brazil, Financing Code 001.

References

[1] Suhartono, Derwin, Wahyu Aditya, Miranty Lestari, and Muhamad Yasin. (2013). "Expert System in Detecting Coffee Plant Diseases." *International Journal of Electrical Energy* **1** (**3**): 156–162.

[2] Pereira, Cassiano Spaziani, Rubens José Guimarães, Edson Ampélio Pozza, and Adriano Alves da Silva. (2008). "Controle de Cercosporiose e da Ferrugem do Cafeeiro com Extrato Etanólico de Própolis" *Ceres* **55(5):** 369-376.

[3] Kushalappa, A.C., M.Akutsu, S.H. Oseguera, G.M.Chaves et al. (1984). "Equations for Predicting the Rate of Coffee Rust Development Based on Net Survival Ratio for Monocyclic Process of Hemileia Vastatrix [Coffea Arabica]." *Fitopatologia Brasileira* (*Brazil).*

[4] Krizhevsky, Alex, Ilya Sutskever, Gepffrey E. Hinton. (2017). "ImageNet Classification with Deep Convolutional Neursal Networks." *Communications of the ACM* **60 (6):** 1097-1105.

[5] Gutte, Vitthal S., Maharudra A.Gitte. (2018). "A Survey on Recognition of Plant Disease with Help of Algorithm". *International Journal of Engineering Science and Computing*, 6(6), 7100-7102.

[6] Mengistu, Abrham Debasu, Dagnachew Melesew Alemayehu, Seffi Gebeyehu Mengistu. (2016). "Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques". *International Journal of Database Theory and Applications*, 9(4), 79-88.

[7] Sladojevic, Srdjan, Marko Arsenovic, Andras Anderla, Dubravko Culibrk, Darko Stefanovic. (2016). "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification." *Computational Intelligence and Neuroscience-Hindawi Publishing Corporation*, V.2016, 1-11.

[8] Fuentes, Alvaro, Sook Yoon, Sang Cheol Kim, Dong Sun Park. (2017). "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition". *Sensors*, 17(9), 1-21.

[9] Haralick, Robert M., K. Shanmugam, Its'Hak Dinstein. (1973). "Textural Features for Image Classification." *IEEE Transactions on Systems, Man, and Cybernetics* **3 (6):** 610-621.

[10] He, Don-Chen, Li Wang. (1990). "Texture Unit, Texture Spectrum, and Texture Analysis." *IEEE Transactions on Geoscience and Remote Sensing* **28 (4):** 509-512.

[11] Mäenpää, Topi. (2003). "The Local Binary Pattern Approach to Texture Analysis: Extensions and Applications.", Academic Dissertation, University of Oulu, Oulun Yliopisto.

[12] Ojala, Timo, Matti Pietikäinen, Topi Mäenpää. (2002). Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns." *Pattern Recognition* **24 (7):** 971-987.

[13] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. (2014). Going Deeper with Convolutions." arXiv preprint, arXiv:1409.4842 **7: 1-14**.

[14] Leun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. (1998). "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86(11) 2278-2324.

[15] Fukushima, Kunihiko. (1980). "A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position". *Biological Cybernetics*, 36(1), 193–202.

[16] Vargas, Ana Caroline Gomes, Aline Paes, Cristina Nader Vasconcelos. (2016) Um Estudo Sobre Redes Neurais Convolucionais e Sua Aplicação em Detecção de Pedestres." *Proceedings of the XXIX Conference on Graphics, Patterns and Images*. **1-4**.

[17] Sousa, Allex de Lima, Marcos Filipe Alves Salame, Firmino José do Nascimento Filho, André Luiz Atroch. (2018). "Redes Neurais Convolucionais Aplicadas ao Processo de Classificação de Cultivares de Guaranazeiros." *XIV Encontro Nacional de Inteligência Artificial e Computacional*. **855-864**.

[18] Cohen, Jacob. (1960)."A Coefficient of Agreement for Nominal Scales – Educational and Psychological Measurement." *Educational and Psychological Measurement* **20** (**1**): 37–46.

[19] Fawcett, Tom. (2006). "An Introduction to ROC Analysis." *Pattern Recognition Letters* **27** (**1**): 861–874.

[20]Patternnet . (2019)."Pattern Recognition Network." Available in: www.mathworks.com/help/deeplearning/ref/patternnet.html.