Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 18, Iss. 2, pp. 27-39, 2020

© Brazilian Computational Intelligence Society

METHODOLOGY FOR CLASSIFYING DISEASES IN PLANTS USING CONVOLUTIONAL NEURAL NETWORKS

Michel Costa
Universidade Federal do Sul e Sudeste do Pará, Marabá, PA, Brazil
michelvictor@unifesspa.edu.br

Vanessa Rezende
Universidade Federal do Sul e Sudeste do Pará, Marabá, PA, Brazil
vanessarezende@unifesspa.edu.br

Cledisson Martins
Universidade Federal do Sul e Sudeste do Pará, Marabá, PA, Brazil
p4tmars@unifesspa.edu.br

Adam Santos
Universidade Federal do Sul e Sudeste do Pará, Marabá, PA, Brazil
adamdreyton@unifesspa.edu.br

Abstract – Convolutional neural networks (CNNs) are one of the deep learning techniques that, due to the computational advance of the last few years, have leveraged the area of computer vision, allowing substantial gains in the most varied classification problems, especially those involving digital images. In this context, this paper aims to propose a methodology for the classification of multiple pathologies related to different plant species. Initially, this methodology involved the image processing and the generation of ten new databases, varying between 50 and 66 classes with greater representation. After training the models (VGG16, RestNet101v1, ResNet101v2, ResNetXt50, and DenseNet169), a comparative study was conducted based on widely used classification metrics, such as test accuracy, *f1-score*, and area under the curve. To attest the significance of the results, Friedman's nonparametric statistical test and two post-hoc procedures were performed, which demonstrated that ResNetXt50 and DenseNet169 obtained superior performances when compared with VGG16 and ResNets.

Keywords – Convolutional neural networks, Plant Diseases, Database, Classification, Deep Learning.

1. INTRODUCTION

The advent of industrialization in the agricultural sector yielded a significant growth in food production worldwide, through the incorporation of technologies that enabled the mechanization of the field and studies that allowed soil fertilization, genetic improvement of seeds and the creation of more effective pesticides [1].

However, even though the mechanization of the countryside has made it possible to increase world food production in the last five decades, the life expectancy of the population has also increased considerably. It is estimated that the food production is expected to grow 70% in the world and double in developing countries to meet the food demand of the world population, which is expected to reach 9.1 billion in 2050, taking into account climate change and rapid urbanization [2].

In this scenario, one of the biggest challenges related to world food security are the diseases that affect plants, as their incidence can result in a drastic reduction in the yield of crops, compromising the quality of fruits, vegetables, grains, and legumes [3] . According to the Society for General Microbiology [4], plant diseases contribute about 10% to 16% of losses in the annual global crop, costing approximately $220 billion.

Usually, these pathologies affect the photosynthetic apparatus and the growth of the plant, in addition to being difficult to control because their populations are variable in time, space, and genotype. About 85% of plant diseases are caused by fungal-like organisms. Other serious plant diseases are caused by bacteria, viruses, and viroids, and few diseases are caused by certain nematodes [3, 5].

In addition, note that these pathologies also have disastrous consequences for small farmers whose livelihoods depend on healthy crops. In most developing countries, more than 80% of agricultural production is generated by small farmers and an estimated loss of income of more than 50% due to pests and diseases, making small farmers a particularly vulnerable group to pathogen-derived disturbances in food supply [6].

In this manner, many efforts have been made to prevent loss of crops due to disease. Historical approaches to widespread application of pesticides have, in the past decade, been increasingly complemented by integrated pest management approaches [6].

However, the abundant use of chemicals, such as bactericides, fungicides, and nematicides to control these pathologies has caused adverse effects on the agroecosystem, making it necessary to use effective techniques for early detection of the causative agent for the control of diseases and the sustainability of the agroecosystem, avoiding the unnecessary waste of financial resources and, consequently, obtaining healthier production [5, 7].

In this context, many studies have been carried out to predict incidents and minimize losses through computational intelligence (CI) techniques based on deep learning (DL) and computer vision for pattern recognition in plants disease images databases [6, 8–11].

Among numerous neural network architectures used in the DL area, convolutional neural networks (CNN) have been established as a dominant approach to the recognition of visual objects due to their satisfactory results in different problems involving computer vision [3, 12, 13]. The application of these models has been driven by the current advances in the processing capacity of computers, mainly in relation to the use of graphic processing units (GPUs).

The rise in the use of these networks is due to the fact that they have proven to reach more accurate results than traditional object recognition methods in a wide range of applications. Besides, the use of CNNs reduces the complexity regarding the definition of the architecture type and allows one to employ relatively deeper models with lower computational cost when compared to traditional methods. Therefore, CNNs have been consolidated as a powerful tool for extracting nonlinear, discriminating, and invariant features [14].

In this context, this paper establishes a methodology for the classification of diseases in plants, which involves image preprocessing and application of different CNN architectures to classify between 50 and 66 diseases of multiple plant species present in the XDB plant disease database, made available by Embrapa [7]. Ten databases were created from the XDB database, with different numbers of classes and distinct image resolutions. The evaluation of results was based on widely used metrics, such as test accuracy stage, f1-score, and area under the curve (AUC). Furthermore, a non-parametric statistical study was carried out to verify the significance of the results, having as input the first metric mentioned above.

The main contributions of this reasearch are:

- A methodological proposal for the classification of diseases in plants based on the processing of digital images from many plants with different types of diseases, which can be useful for specialists in this field, enabling early diagnosis that contributes to the reduction of profit losses in the agricultural sector;
- A tool to control the spread of epidemics, contributing to food security in food production;
- A method for image preprocessing which facilitates the creation of standard databases from plant diseases image repositories that are still under development and have images with varied quality.
- A comparative study between CNNs, which proved the significance of the results obtained for different databases with multiple plant diseases, from different Brazilian regions.

The remainder of this paper is organized as follows. In section 2, the state-of-the-art works are revisited. Section 3 presents the proposed methodology, along with the database and model settings. Section 4 highlights the results and analysis. Section 5 synthesizes the conclusions and some future directions.

2 RELATED WORK

CNNs are one of the most used architectures in the supervised DL field, acting such that the data characteristics can be exploited automatically to obtain the best results. This type of network can be summarized as a model based on the integration of biological vision with the neural system [5]. These networks have high power of precision in classification tasks and object recognition [3].

Recent applications of CNNs involve the detection of pedestrians and road signs [15], image recognition [16], tumor detection [17, 18], genus identification of small insects [19], and detection of diseases in plants [9–11].

Plants infected with diseases usually exhibit visible marks or lesions on either the leaves, stems, flowers, and/or fruits. Generally, each disease or pest condition presents a unique visible pattern, which can be used to uniquely diagnose the anomaly [20]. Based on this assumption, many works have been developed based on CNNs for detecting plant diseases. This section comprises an overview of the literature work in the field of disease recognition in plants through different CNN models.

In [21], it was pointed out the importance of the early diagnostic and precise identification of apple leaves diseases, which might help in controlling the propagation of pathologies and ensure the healthy development. The research proposed an approach to identify diseases in the apple leaves with a CNN based on the AlexNet model, using a database composed of 13,689 images. The results showed that the AlexNet variant obtained a general precision of 97.62%, with a significant reduction in the number of parameters. In terms of precision, an improvement of 10.83% over the original AlexNet was obtained.

In [11], the work covered the development of a model for plant diseases recognition based on the classification of leaves images using a traditional pre-trained CNN. All the collected images were downloaded from the Internet and searched by diseases and plant names in many sources and languages. The images were grouped in 15 different classes, where 13 of them represent plant diseases which could be visually defined by the leaves. For the precision test, the cross validation technique with ten repetitions was used to assess a predictive model. The experimental results of this model obtained the precision between 91% and 98%.

In the field of mobile applications, an automated image recognition system with the use of smartphones and high-performance processors to detect plant diseases was implemented in [6]. For this task, they used CNNs, such as the AlexNet and GoogleNet architectures, for comparative purposes. As expected, GoogleNet performed consistently better than AlexNet. In one of the test cases, the first achieved an overall accuracy of 99.35% in the detection of 26 diseases in 14 plant species, in a total of 54,306 color images.

As noted, CNNs have been proved their potential when applied to the classification tasks. Besides, their power have demonstrated that they can exceed human performance in some object recognition tasks [16]. However, for the success of these models, large amount of training data labeled with the respective class is crucial to produce a rich feature hierarchy.

A method for identifying soy diseases, based on a transfer learning approach and pre-trained AlexNet and GoogleNet architectures, was developed in [22]. To identify three diseases, these models were trained using 649 images of unhealthy soy leaves and 550 image of healthy soy leaves. In addition, a five-fold cross-validation strategy was used. The models based on AlexNet and GoogleNet reached an accuracy of 98.75% and 96.25%, respectively, which demonstrated the efficiency for identifying soybean diseases.

Regarding this proposed work, other studies have already demonstrated the feasibility of research in the XDB database [23] [24]. The previous works highlighted some intermediate results, which were concentrated in the comparison studies between the different architectures. The refinement of this idea resulted in the methodology for the classification of plant diseases described in detail in the following section.

3. PROPOSED METHODOLOGY

This research aims to develop a methodology based on the image preprocessing and the application of different CNN models to perform the classification of plant diseases from digital images encompassed in the XDB plant disease database. This database was originally created and made public available by the Brazilian Agricultural Research Corporation (EMBRAPA) [7]. To achieve this goal, a methodology based on four phases was established, as presented schematically in Figure 1.

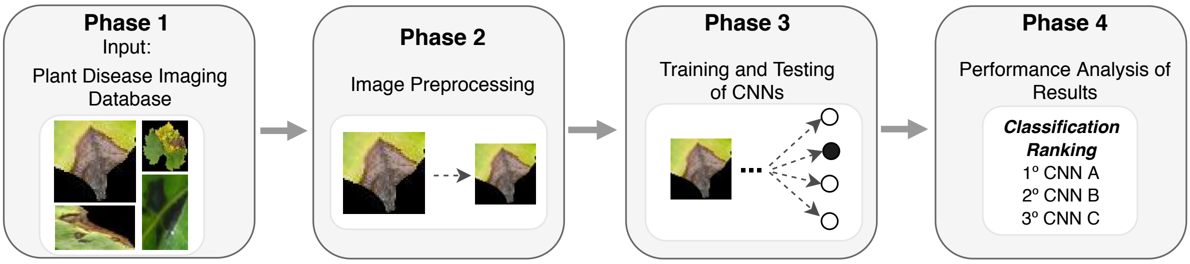

Figure 1: Methodology for the classification of plant diseases.

The first phase of this methodology involved the selection of the database with diseases that affect plants as an input for the CNN that were used. However, to make the training of the CNNs possible, it was necessary to include an intermediate phase, called image preprocessing, which can standardize the size of all images and eliminate images with extremely small dimensions. After this standardization, the training and testing phases of the CNNs began, which performed the extraction of relevant features that made it possible to classify the pathologies. Finally, the performance analysis of the results was carried out, which aimed to evaluate the output of the applied CNNs and to classify the best networks in a ranking based on non-parametric statistical tests.

3.1 PLANT DISEASE DATABASES

The database used in this study was created by researchers from EMBRAPA. They collected images of plant pathologies and the organization of the database was called Plant Disease Data Base (PDDB), which is available at https://www.digipathos-rep.cnptia.embrapa.br/. As a result, 2,326 images from 171 diseases and other disorders were gathered, comprising 21 species of plants [7].

However, as mentioned by the researchers, the PDDB database does not have a number of images that make it possible to apply DL techniques for the extraction of disease features, as most classes are not representative [7]. Based on this issue, to increase the size of the database, the images were divided into smaller images containing individual lesions or regions of symptoms, according to some criteria, such as the extraction of leaves from diseased plants [7]. Thereby, this expanded database (XDB) currently contains 46,513 images and is the object of study for this work. Figure 2 depicts some images from the XDB database.

3.2 IMAGE PROCESSING AND DATABASE ORGANIZATION

Before using the images from the XDB database as input to the CNNs, it was necessary to carry out preprocessing steps, because to execute DL models, it is crucial that the images present consistency regarding the position and size of the region of interest, and sufficient representativeness in each class such that pathologies can be properly classified, improving the training and, subsequently, the testing of CNNs.

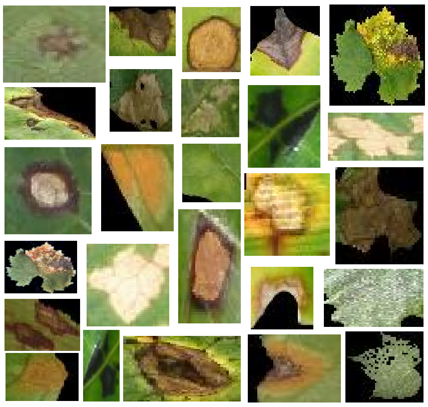

Figure 2: Sample images from the XDB.

The image preprocessing stages intend to create databases with different features based on the images from the XDB, for testing the CNN models under different settings. Another motivation for carrying out these stages were the need of a method that could facilitate the elimination of very small images that could hinder the generalization of CNN models. Preprocessing involved two main steps: i) image selection and (ii) image resizing.

The step of selecting the images basically consisted of a validation that is justified by the fact that the XDB database is composed of images with different dimensions, being some of them extremely small. In this sense, a method was implemented to perform the selection of the desired images for the creation of ten databases that were used in this paper.

This validation was based on a proximity factor to the target dimensions, called the difference rate, which considered the input dimensions, comparing the size of the current image with the target size. If the current image has dimensions below the minimum limit defined by the rate of difference, it is eliminated from the group of images that compose the new database.

Table 1 shows the values corresponding to the rates applied to the dimensions established for the images comprised in the databases created in this paper.

Table 1: Difference rate applied according to the chosen dimensions.

Dimensions	Difference Rate
32×32	0
40×40	5%
48×48	10%
56×56	15%
64×64	20%

When one disregards the difference rate in the creation of databases with 32×32 images, all images equal to or greater than these dimensions were considered. This methodology was adopted because the applied CNNs assume 32×32 as the minimum image dimensions for their input. Although it is possible to define a rate of difference and then resize the images to the desired target dimensions, it was observed that the images with dimensions smaller than 32×32 were extremely small.

From the dimensions 40×40 to 64×64, the difference rate varied with an increase of 5% as the target dimensions were increased, i.e., all images that had a dimensions starting from $d_{px \times px} - (d_{px \times px} \times t)$ would be considered for a given new database, where $d_{px \times px}$ refers to the target dimensions and t is the difference rate.

Afterwards, the selected images proceeded to the resizing step, which standardizes the dimensions accepted in the selection step to the target dimensions. Figure 3 presents examples from the preprocessing steps.

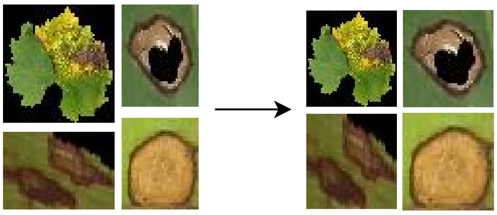

Figure 3: Examples of images before and after the preprocessing steps.

After the preprocessing steps, ten databases were generated with configurations that can be observed in Table 2. Initially, the databases were created with images that have dimensions varying by eight pixels, i.e., dimensions starting from 32×32 to

64×64. These first five databases encompassed the 50 most representative classes, named in this study with the suffix "final". Besides, more five databases were created with a number of classes varying according to the most representative classes per target dimensions, starting from 60 classes in the database of 64×64 images and reaching up to 66 classes in the database of 32×32 images. These last five databases followed the same previous pattern in dimensions and were classified using the suffix "hard".

Table 2: Databases' configurations.

Designation	Target dim.	#classes	#images
bd32final	32×32	50	35,430
bd32hard	32×32	66	36,558
bd40final	40×40	50	32,043
bd40hard	40×40	65	33,049
bd48final	48×48	50	29,020
bd48hard	48×48	64	29,878
bd56final	56×56	50	27,079
bd56hard	56×56	62	27,786
bd64final	64×64	50	25,211
bd64hard	64×64	60	25,778

In this study, the representativeness in each class is related to the largest number of images belonging to a given class, with no distinction of belonging to the same culture or pathology, as the XDB is not a balanced database in its original form.

In addition, due to the fact that the images in the XDB are small, which can more effectively represent the disease, the databases of images with dimensions 32×32 have a larger number of images, as expected. On the other hand, although with the largest resolution, the databases with images of dimensions 64×64 are the ones that group the smallest number of images.

3.3 TRAINING AND TESTING OF CNNs

3.3.1 MODELS' CONFIGURATION

In this study, the VGG model was selected because it is a reference in the area of computer vision, being one of the first to obtain satisfactory results in a large volume of data. The DenseNet and ResNet models were selected because they have superior results in the current literature. ResNetXt was chosen because it concentrates characteristics from VGG and ResNet models, making it interesting to compare the performances between these CNNs. Specifically, the VGG16, DenseNet169, ResNet101v1, ResNet101v2, and ResNetXt were adapted and then applied to the ten databases generated from XDB (Table 2).

In terms of adaptations to the original architectures, the VGG16 model was configured with 200 neurons in the first two fully connected layers, both followed by a dropout regularization of 0.5, before the output layer. The DenseNet model had the addition of a fully connected layer composed of 200 neurons, followed by a dropout regularization of 0.5, before the output layer. The last configuration was adopted for the ResNet101v1, ResNet101v2, and ResNetXt models. All CNNs were pre-trained in the ImageNet [25] database and their implementations were based on the Keras framework [26].

The training of all models was performed using the stochastic gradient descent (SGD) algorithm [27] with momentum equal to 0.9, and learning rate starting at 10^{-3} and decreasing to a minimum of 0.5×10^{-6}. For all CNNs, 200 epochs and batch size of 64 were considered, with a reduction in the learning rate by the factors 10^{-1}, 10^{-2}, 10^{-3}, and 0.5×10^{-1} when the number of epochs was greater than 80, 120, 160, and 180, respectively, considering that the result in the test data did not improve after five consecutive epochs.

The operation of the SGD involves performing one update at a time, which generally makes the training process much faster. In general, the SGD makes it possible to jump to new and potentially better local minimum [27]. It has been shown that when the learning rate is slowly decreased, the algorithm demonstrates the same convergence behavior as the batch descending gradient, approximately converging to a local or global minimum for convex and non-convex optimizations, respectively [28]. For these reasons, the reduction in the learning rate over time is justified in this paper.

3.3.2 DATA SEPARATION

When one applies the CNN models, generally three data sets are required: training, validation, and testing. The training set is the collection of images to be used by the network to automatically learn its intrinsic parameters (e.g., the weights). The validation set is used to adjust hyperparameters that are essential for fine-tuning purposes. These hyperparameters include, among others, learning rate, batch size, number of epochs, and network architecture. The values of these hyperparameters are often defined empirically, as they are linked to the problem [3].

Therefore, the model being trained can be evaluated in the validation set at the end of each epoch, which allows the training process to be monitored and possible underfitting/overfitting to be detected. Generally, training and validation sets come from the same data source that is subdivided.

The testing data set is used only when the training phase is completed, with the aim of evaluating the final generalization of the model. The accuracy in the testing set is, therefore, one of the most important metrics to be computed, as it provides an overview of the model's performance beyond the process of exploring the hyperparameters.

Specifically for this paper, there is no distinction between the validation and testing data sets, as the hyperparameters are not optimized along with the validation set. The majority of the hyperparameters were fixed or varied empirically from works in the literature. Thus, for each database (Table 2), 70% of the images were used in the training stage, while the remaining images were employed for evaluating (validation/testing) the learned models. Note that the training and testing sets were shuffled at each epoch for a given execution.

3.3.3 REGULARIZATION TECHNIQUES

This work applied regularization techniques widely used to improve the generalization of the CNN models and reduce the overfitting. Generally, the first regularization step involves gathering more data, that is, increasing the amount of images of pathologies in plants. However, obtaining many images for a given class can be complicated in a context of plant diseases, especially because the XDB still in the construction process.

As a result, the proposed methodology used data augmentation techniques based on images that already existed in the training data. These techniques consist of transforming the geometry or intensity of the original images to make them as new images, by employing some operations, such as rotation, mirroring, random cropping, zooming, adding noise, changing the contrast and brightness values, and so on. The augmentation operations can be performed in different manners, using one or more transformations per image. Besides, the transformations can be also applied before the start of the entire training process or when each batch of images is loaded.

In this paper, an automatic method was developed to increase the number of images in the training data. This method applied horizontal rotation and displacement, vertical displacement, and ZCA whitening [29,30], which were performed when each batch of images was loaded during each training epoch. All data augmentation effects were performed using the Keras framework.

Among the regularization techniques applied directly to the models, the dropout can be considered, which is a mechanism that eliminates a random set of neuron activations to avoid excessive adjustment during the training stage, such that each neuron can learn to detect a useful feature to produce the correct response. In this research, in all CNNs, following each fully connected layer (except the output layer), a dropout equivalent to 0.5 was applied.

4. RESULTS AND ANALYSIS

The strategy used in this paper involved starting to train these algorithms from an intermediate point, instead of performing the training from the beginning. This is because the ImageNet database has images of plants [25], even in healthy condition, thus the pre-training in this database contributed to a greater efficiency in the refined training carried out after this stage. This fact also justifies the use of the SGD algorithm and the decay in the learning rate over epochs, as the objective is to slowly optimize pre-trained models as an refinement, without abruptly exploring the space of features.

It should be noted that all experiments were performed on an NVIDIA GeForce GTX 1050 TI GPU with 4 GB of video memory, along with a 2.2 GHz (4.1 GHz) Intel Core i7-8750H processor and 16 GB of RAM. The average computational time of the most costly executions considering the database "32hard", which has the largest number of images, were ResNet101v1 and ResNet101v2, with 10 hours and 10 min, and 11 hours and 30 min, respectively, followed by DenseNet with approximately 9 hours and 30 min and ResNetXt with 6 hours. The CNN with the lowest average execution time was VGG16, which lasted 3 hours and 30min. Note that although the CNNs were executed in the GPU, making more than ten executions for each model proved to be costly in time. Considering all the executions, the averaged times presented relatively low standard deviations.

4.1 PERFORMANCE EVALUATION THROUGH ACCURACY, *F1-SCORE*, AND AUC

Considering a shuffle strategy in the database split during each epoch from the training process, from ten executions of each CNN in each database, the performance averages for the accuracy metrics, *f1-score*, and AUC in the test data are presented in Tables 3, 4 and 5, respectively. In a global analysis, it is possible to verify that, in most cases, ResNetXt50 maintained a better performance in databases that comprised images with dimensions greater than 40 × 40, among all CNNs and metrics considered, while DenseNet169 obtained better results in databases that involved images with smaller dimensions than 48 × 48 and results marginally lower than the ResNetXt50 for databases that included images with dimensions larger than 40 × 40. On the other hand, ResNet101v1 obtained the worst results in the databases of images with dimensions 32 × 32, although it surpassed ResNet101v2 in all databases of images with dimensions greater than 32 × 32. ResNet101v1 also had results superior to those reached by VGG16, in databases of images with dimensions larger than 48 × 48, except for a tie performance in the database "56final". The metrics related to the test accuracy and AUC better highlighted the performances of the CNNs, showing relevant differences between the achieved results.

When one considers the results based on the accuracy metric in the test (Table 3) in databases of images with dimensions 40 × 40 and 64 × 64, it is possible to observe that the performances of ResNetXt50 and DenseNet169 remained similar, with small advantage for the former. A slightly larger difference between them can be seen in the databases of images with dimensions 48 × 48 and 56 × 56, with a positive highlight for ResNetXt50, while in the databases of images with dimensions 32 × 32, DenseNet169 obtained the best results among all CNNs. On the other hand, the ResNets presented a lower performance, with

ResNet101v2 being the negative highlight for all databases used in this experiment, while ResNet101v1 still managed to obtain results that were marginally superior to those achieved by VGG16.

Table 3: Average of test accuracy for the databases.

	VGG16	**ResNet101v1**	**ResNet101v2**	**ResNetXt50**	**DenseNet169**
32final	0.936	0.876	0.916	0.940	**0.942**
32hard	0.929	0.878	0.911	0.936	**0.940**
40final	0.947	0.943	0.933	**0.955**	0.954
40hard	0.941	0.939	0.930	**0.950**	0.949
48final	0.955	0.953	0.943	**0.965**	0.962
48hard	0.950	0.946	0.938	**0.961**	0.957
56final	0.959	0.959	0.951	**0.966**	0.962
56hard	0.954	0.958	0.949	**0.965**	0.961
64final	0.963	0.964	0.953	0.970	**0.971**
64hard	0.960	0.966	0.949	**0.968**	0.967

As summarized Table 4, a similar trend in the results was observed when one considers the metric *f1-score* in databases of images with dimensions larger than 40 × 40, in which ResNetXt50 achieved a slightly higher performance than DenseNet169, although the latter achieved the best results in the databases of images with dimensions 32 × 32 among all CNNs. On the other hand, although VGG16 has obtained results relatively superior to those reached by ResNets in the databases of images with dimensions 32 × 32 and in the database "40final", in all other subsequent databases the performance of this CNN was lower when compared to ResNet101v1, although it was considerably better when compared to ResNet101v2.

Table 4: Average of *f1-score* for the databases.

	VGG16	**ResNet101v1**	**ResNet101v2**	**ResNetXt50**	**DenseNet169**
32final	0.917	0.852	0.896	0.925	**0.926**
32hard	0.881	0.828	0.864	0.904	**0.910**
40final	0.930	0.928	0.913	**0.940**	**0.940**
40hard	0.896	0.901	0.885	0.915	**0.916**
48final	0.934	0.937	0.920	**0.950**	0.946
48hard	0.907	0.910	0.889	**0.931**	0.922
56final	0.940	0.940	0.931	**0.950**	0.948
56hard	0.911	0.926	0.907	**0.938**	0.930
64final	0.944	0.948	0.930	**0.958**	0.956
64hard	0.929	0.936	0.913	**0.946**	**0.946**

As synthesized in Table 5, the results pointed out by the AUC metric reinforced that the best results were from ResNetXt50 followed by DenseNet169 by a marginal difference, where it is possible to highlight a tie in the results for database "40hard" and an advantage of DenseNet169 in the databases of images with dimensions 32 × 32. With respect to VGG16, it was demonstrated that its results were superior to those from ResNets in databases of images with dimensions less than 56 × 56, except for the result from the database "40hard", where the performance of ResNet101v1 was marginally higher than VGG16. In the database "56final", there was a tie between the performance of ResNet101v1 and VGG16, and in the database "56hard" and in the two databases of images with dimensions 64 × 64, ResNet101v1 surpassed the performance of VGG16 and ResNet101v2. Moreover, for the databases of images with dimensions 32 × 32, the AUC metric evidenced that ResNet101v2 obtained better results than ResNet101v1, although they were not sufficient to surpass the results of the best CNNs, i.e., ResNetXt and DenseNet169.

From the results presented by accuracy, *f1-score*, and AUC metrics, it was verified that the ResNetXt50 and DenseNet169 models had a better performance in terms of all databases used in this study, with a slight advantage for ResNetXt50.

It is possible to infer that one of the reasons that made ResNetXt50 obtained a superior result in comparison to the other models is due to the simplicity of its architecture, as it is highly modular and works with the reuse of residual blocks, which reduce the computational cost in the executions. Furthermore, ResNetXt50 brought together the strong points from VGG and ResNet, such as the increase in the dimension of cardinality, which has proven to have satisfactory results in the literature [31]. For the results of DenseNet169, it can be deduced that they were satisfactory due to the advanced information flow that travels through the network during the training stage, making all layers of the model aware of the information inherent in the previous layers. Another point to consider is the fact that both ResNetXt50 and DenseNet169 have regularization mechanisms that significantly reduce problems related to the vanishing gradient, which is a challenge in the VGG16 architecture. For the results obtained by ResNets, it is assumed that the pre-training in the Imagenet database did not favor or was not sufficient for a better

Table 5: Average of AUC metric for the databases.

	VGG16	**ResNet101v1**	**ResNet101v2**	**ResNetXt50**	**DenseNet169**
32final	0.957	0.922	0.945	0.960	**0.962**
32hard	0.937	0.907	0.925	0.947	**0.951**
40final	0.963	0.962	0.954	**0.969**	0.968
40hard	0.945	0.946	0.937	**0.953**	**0.953**
48final	0.966	0.965	0.958	**0.972**	0.970
48hard	0.950	0.948	0.938	**0.959**	0.955
56final	0.969	0.969	0.964	**0.974**	0.971
56hard	0.953	0.958	0.949	**0.965**	0.960
64final	0.969	0.970	0.961	**0.975**	**0.975**
64hard	0.963	0.965	0.951	0.969	**0.970**

classification performance given the fact that the pre-training included only images of healthy plants, or even for the ResNet architectures, which have great depth, needing more epochs to achieve more acceptable results.

For the purpose of visualizing the performances of CNNs in the training and testing stages, the following results are presented for the 32hard and 64hard databases, as they denote extremes in terms of image quality. Note that ResNet101v1 was excluded from this analysis to make the result visualization easier.

In Figure 4, the performances of the applied CNNs are presented in terms of accuracy throughout the epochs of the training stage in the database “32hard”. It is observed that ResNetXt50 and DenseNet169 have already started the training with results superior to the other CNNs, although the first one has a slight advantage until around the epoch 80. VGG16 and ResNet101v2 initially maintained performances with some differences until around the epoch 60, where up to approximately the epoch 80, the performances were significantly similar. In the subsequent epochs, VGG16 surpassed ResNet101v2 until the end of training.

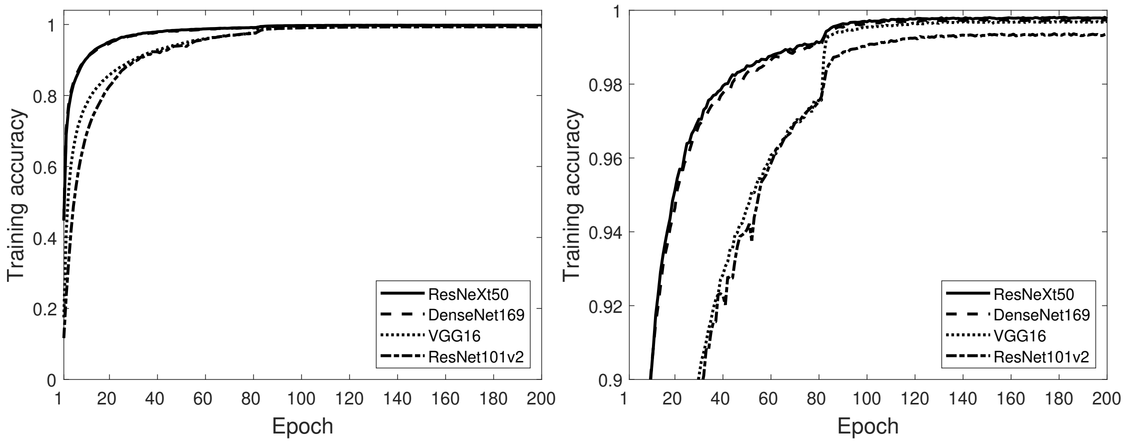

Figure 4: Performance of ResNetXt50, DenseNet169, VGG16, and ResNet101v2 in terms of accuracy during the training stage in the database “32hard”.

With regard to the performance of CNNs in the testing stage on the database “32hard”, it is verified in Figure 5 that the best results of ResNetXt50 and DenseNet169 in the training contributed positively to a better generalization in the classification of new images from the various classes of plant diseases, while a relatively unsatisfactory performance was noticeable in relation to ResNet101v2, which does not characterize a *overfitting* effect. VGG16 obtained an intermediate performance, approaching the best results from the epoch 80, where it is possible to perceive the positive effect of applying a reduction in the learning rate for this model, as well as the major distance between its results and the ones obtained by ResNet101v2.

Figure 6 shows the performances of CNNs in terms of accuracy throughout the epochs of the training stage in the database “64hard”. It is inferred again that only ResNetXt and DenseNet169 started the training stage with superior results, while after the epoch 100, the performance among all CNNs remained similar, which demonstrated that the strategy of generating new images during the training had a positive effect along the epochs for all CNNs applied in this database. Specifically, despite minor differences from the epoch 100, ResNetXt50 remained with a superior overall performance, while ResNet101v2 remained in last place.

In the testing stage, on the database “64hard”, some discrepancies were more evident in the results, when compared to the training results, as can be seen in Figure 7. An almost imperceptible difference between the performance of ResNetXt50 and DenseNet169 is noted, in contrast to the performance of the VGG16, where the difference between it and the other CNNs is clear. Although the training results demonstrate similarities, the ability of CNNs to generalize to new test images demonstrated

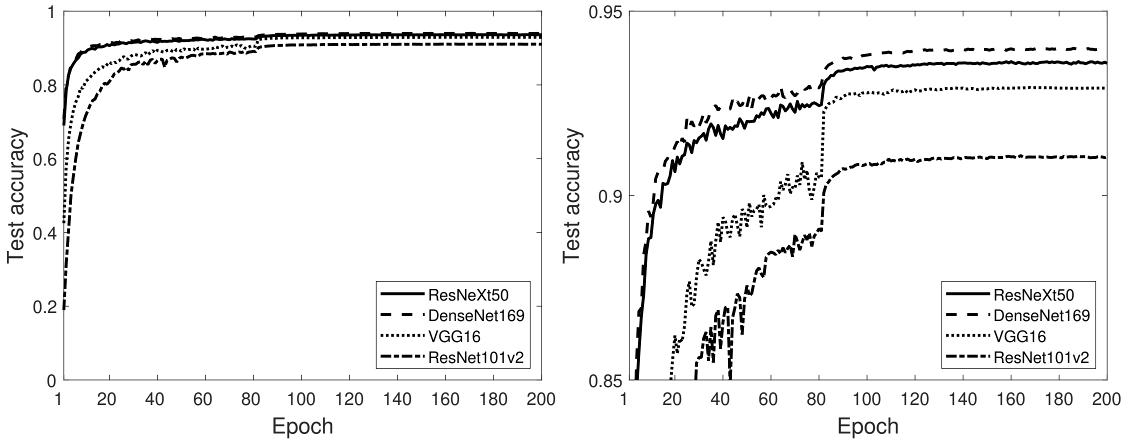

Figure 5: Performance of CNNs ResNetXt50, DenseNet169, VGG16, and ResNet101v2 in terms of accuracy during the testing stage in the database “32hard”.

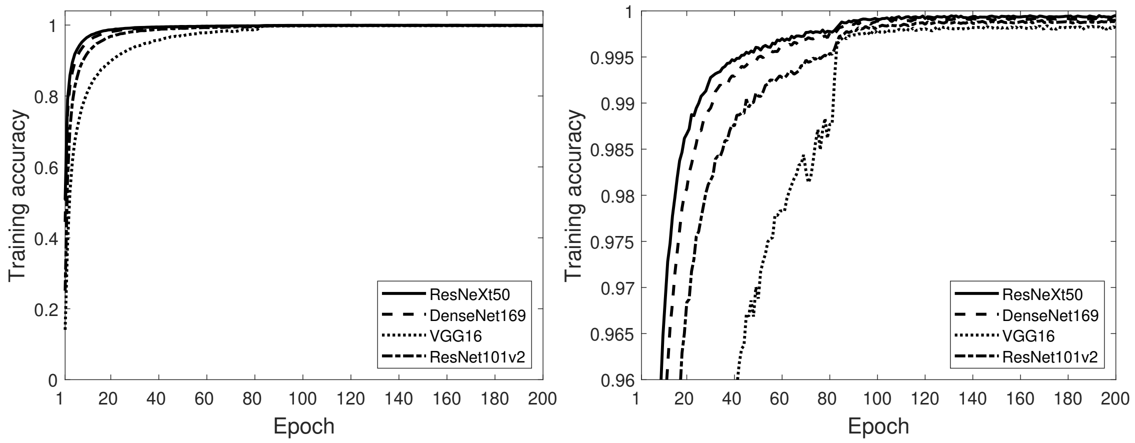

Figure 6: Performance of ResNetXt50, DenseNet169, VGG16, and ResNet101v2 in terms of accuracy during the training stage in the database “64hard”.

that the trained model of ResNet101v2 had some specialization in the training data, when compared with the other CNNs.

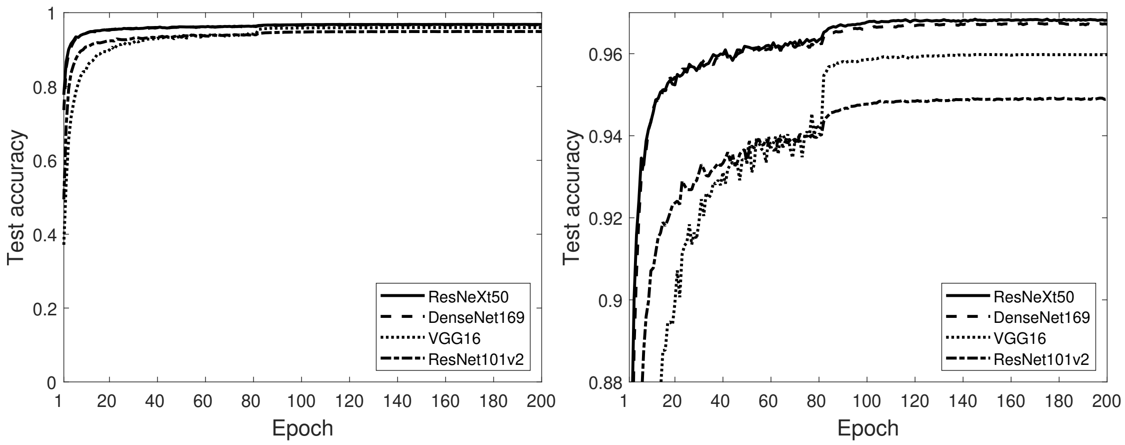

Figure 7: Performance of the ResNetXt50, DenseNet169, VGG16 and ResNet101v2 in terms of accuracy during the testing stage in the database “64hard”.

4.2 NON-PARAMETRIC STATISTICAL TEST

To verify the significance in all possible peer-to-peer comparisons between CNNs, Friedman’s non-parametric statistical test was performed, complemented by the post-hoc procedures Shaffer and Bergmann to adjust the initially computed p value. These experiments assumed the following null and alternative hypotheses, respectively: H_0 which considers that the results of the compared algorithms have no differences; and H_1 which considers that the results of the compared algorithms differ.

To apply the Friedman test, the original results in terms of test accuracy (Table 3) were converted using a ranking-based transformation [32]. Table 6 summarizes the computed ranking from Friedman test, which highlights the ResNetXt50 and DenseNet169 models as the best ones, with a slight advantage for the former, while ResNet101v2 was considered the worst CNN taking into account all databases of this study. Therefore, it is possible to infer that there is significance in the multiple possible comparisons between CNNs, and in some cases the hypothesis H_0 can be refuted and the hypothesis H_1 can be accepted.

Table 6: Average ranking computed by the Friedman test.

CNN	Ranking
ResNetXt	1.3
DenseNet169	1.7
VGG16	3.4
ResNet101v1	3.8
ResNet101v2	4.8

Table 7 presents all hypotheses for comparisons and the p values unadjusted and adjusted by the Shaffer and Bergmann *post-hoc* procedures. It is important to note that the lower the value of p, the greater the significance between the results of the compared CNNs, which culminates in the rejection of the null hypothesis H_0 and, consequently, the acceptance of the alternative hypothesis H_1 with a high degree of confidence.

Thus, as evidenced by Table 7, ResNetXt50 has a significantly superior performance when compared to ResNet101v1, ResNet101v2, and VGG16. On the other hand, the comparison between ResNetXt50 and DenseNet169 did not demonstrate adequate significance to guarantee any inference of better or worse performance with a satisfactory degree of confidence. When one compares the performance of VGG16 with ResNet101v1, it is noticed that there is no significance, while the comparison between VGG16 and ResNet101v2 suggests some significance, where the results presented by VGG16 were superior. Finally, the hypothesis that made the comparison between the performances of ResNets was not significant.

Table 7: Unadjusted (Friedman) and adjusted (Shaffer and Bergmann) p values for multiple comparisons.

Hypotheses	p ***unadjusted***	p **Shaffer**	p **Bergmann**
ResNet101v2 vs. ResNetXt50	7.430983723703081E-7	7.430983723703081E-6	7.430983723703081E-6
ResNet101v2 vs. DenseNet169	1.1648657367238044E-5	6.989194420342827E-5	6.989194420342827E-5
ResNet101v1 vs. ResNetXt50	4.0695201744492504E-4	0.0024417121046695503	0.0024417121046695503
VGG16 vs. ResNetXt50	0.0029794666563330327	0.017876799937998195	0.01191786662533213
ResNet101v1 vs. DenseNet169	0.0029794666563330444	0.017876799937998268	0.01191786662533213
VGG16 vs. DenseNet169	0.01620954140922542	0.06483816563690167	0.03241908281845084
VGG16 vs. ResNet101v2	0.04771488023735128	0.19085952094940511	0.19085952094940511
ResNet101v1 vs. ResNet101v2	0.1572992070502848	0.47189762115085443	0.3145984141005696
ResNetXt50 vs. DenseNet169	0.5716076449533316	1.1432152899066632	1.1432152899066632
VGG16 vs. ResNet101v1	0.5716076449533326	1.1432152899066632	1.1432152899066632

In addition, a non-parametric statistical test of paired comparisons (1 × N) called contrast estimation based on medians was performed [33], intended to estimate the contrast between the medians of the results, considering all peer-to-peer comparisons between CNNs. This estimation is a quantitative difference that is especially useful for computing the extent to which one CNN outperforms the other. The performance of CNNs is reflected by the magnitudes of the differences between them in each domain.

Table 8 synthesizes the results of applying the contrast estimation test on the results related to the accuracy metric in the testing step (Table 3). Note that comparisons between the same models of CNNs have no value precisely because they are not valid comparisons. In comparisons between different CNNs, this test demonstrated that the ResNets did not outperform any other CNNs, and ResNet101v1 outperformed ResNet101v2 by a small margin.

In turn, comparisons involving the performance of VGG16 demonstrated that it surpassed only ResNets, with larger and lower differences to ResNet101v2 and ResNet101v1, respectively. Among the best results, ResNetXt50 surpassed the performance of all other CNNs.

In summary, the statistical tests validated many conclusions obtained in the experiments, such as the fact that the ResNetXt50 and DenseNet169 models obtained the best results when compared to the others, and the low generalization performance of the ResNets. In addition, these tests refuted significance in the following comparisons: ResNet101v1 vs. ResNet101v2; ResNetXt50 vs. DenseNet169; and VGG16 vs. ResNet101v1.

5 CONCLUSION

The present work proposed a methodology for classifying diseases in plants based on the application of DL techniques in a database of digital images from multiple pathologies referring to different plants. To achieve the proposed objectives, the

Table 8: Non-parametric statistical test as contrast estimation.

	VGG16	**ResNet101v1**	**ResNet101v2**	**ResNetXt**	**DenseNet169**
VGG16	0.000	0.002410	0.01146	-0.008925	-0.007413
ResNet101v1	-0.002410	0.000	0.009048	-0.01134	-0.009823
ResNet101v2	-0.01146	-0.009048	0.000	-0.02038	-0.01887
ResNetXt50	0.008925	0.01134	0.02038	0.000	0.001512
DenseNet169	0.007413	0.009823	0.01887	-0.001512	0.000

images from the XDB database were subjected to preprocessing techniques based on the difference rate and resizing, aiming to standardize the images that served as input for the training of CNNs. It should also be noted that ten databases of different dimensions were created, based on the XDB database, to assess the performances of CNNs when submitted to different situations. The databases involved examples of different classes, taking into account the most representative classes.

After training the models, the evaluation of results took place through the metrics accuracy, *f1-score*, and AUC. Then, to assess the performances of the CNNs used in the classification task and generate a ranking of performances, Friedman's non-parametric statistical study was performed considering the *post-hoc* procedures Shaffer and Bergmann.

From the results, ResNetXt50 achieved promising classification results when compared to those obtained by VGG16, ResNet101v1, and ResNet101v2, in terms of test accuracy and other related metrics, Friedman's non-parametric statistical test, and contrast estimation test. On the other hand, taking into account the Friedman test, the comparison between the performances of ResNetXt50 and DenseNet169 was not significant enough to infer which CNN had the best performance. Differently, the contrast estimation test showed a small difference between them, revealing the best performance of ResNetXt50.

The use of CNNs proved to be efficient with regard to the task of classifying images of diseased plants, as well as the configuration selected for training, except for the ResNets which could not obtain efficient results as the other models tested under the same conditions. Furthermore, it is believed that the increase in the number of epochs could improve the classification accuracy for these CNNs, although it is still necessary to evaluate the cost-benefit in relation to the computational cost.

In general, although the proposed methodology in this paper is an improvement on the work proposed in [23], the present work includes more CNNs architectures and databases with different sizes and challenges, with the objectives of validating the methodology and assessing the distinct scenarios provided through the aforementioned inclusions.

Moreover, the experiments showed that the selected CNNs are able to learn the patterns related to the pathologies present in the digital images of plants, being useful to classify problems with a varied amount of data and multiple classes in a satisfactory manner.

To continue this research, the ideas for possible future work are:

- The creation of other databases of plant diseases based on the XDB database with dimensions greater than 64 × 64;
- Change the configuration of CNNs, increasing the number of epochs, to assess the impact on the performance of CNNs, especially ResNets;
- Use the same methodology in more databases of plants diseases that are still under construction to test the generalization of the proposed methodology;
- Combine two or more criteria for decision making about the best CNN model;
- Add other non-parametric statistical tests in the performance analysis stage, with the aim of reinforcing the reliability of the results, as well as add other performance metrics as input for these tests.

REFERENCES

[1] V. Thorstensen, V. D. R. Gabriel and A. L. Mesquita. "A participação de Brasil e estados unidos na formulação das regras multilaterais do comércio agrícola". *Revista de Direito Internacional*, vol. 15, no. 3, March 2019.

[2] M. Z. Ms. Ann Steensland. "2018 Global Agricultural Productivity Report, Agriculture for a Healthy Sustainable World". Technical Report (GAP Report®), Global Harvest Initiative, October 2018.

[3] Y. Toda and F. Okura. "How Convolutional Neural Networks Diagnose Plant Disease". *Plant Phenomics*, vol. 2019, pp. 1–14, March 2019.

[4] S.G.M. "Combating plant diseases is key for sustainable crops". Available in `https://www.sciencedaily.com/releases/2011/04/110411194819.html`, accessed in 2020/07/09.

[5] K. Golhani, S. K. Balasundram, G. Vadamalai and B. Pradhan. "A review of neural networks in plant disease detection using hyperspectral data". *Information Processing in Agriculture*, vol. 5, no. 3, pp. 354–371, September 2018.

[6] S. P. Mohanty, D. P. Hughes and M. Salathé. "Using Deep Learning for Image-Based Plant Disease Detection". *Frontiers in Plant Science*, vol. 7, September 2016.

[7] J. G. A. Barbedo, L. V. Koenigkan, B. A. Halfeld-Vieira, R. V. Costa, K. L. Nechet, C. V. Godoy, M. L. Junior, F. R. A. Patricio, V. Talamini, L. G. Chitarra, S. A. S. Oliveira, A. K. N. Ishida, J. M. C. Fernandes, T. T. Santos, F. R. Cavalcanti, D. Terao and F. Angelotti. "Annotated Plant Pathology Databases for Image-Based Detection and Recognition of Diseases". *IEEE Latin America Transactions*, vol. 16, no. 6, pp. 1749–1757, June 2018.

[8] R. Gandhi, S. Nimbalkar, N. Yelamanchili and S. Ponkshe. "Plant disease detection using CNNs and GANs as an augmentative approach". In *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*. IEEE, May 2018.

[9] B. Liu, Y. Zhang, D. He and Y. Li. "Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks". *Symmetry*, vol. 10, no. 1, pp. 11, December 2017.

[10] A. dos Santos Ferreira. "Redes Neurais Convolucionais Profundas na Detecção de Plantas Daninhas em Lavoura de Soja". Master's thesis, Universidade Federal de Mato Grosso do Sul, Campo Grande, 2017.

[11] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk and D. Stefanovic. "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification". *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–11, 2016.

[12] R. Oliveira, E. Alves and C. Malqui. "Redes Neurais Convolucionais Aplicadas à Preensão Robótica". In *Anais do 13 Congresso Brasileiro de Inteligência Computacional*, Curitiba, PR, January 2018. ABRICOM.

[13] D. de Paiva Penha. "Rede Neural Convolucional Aplicada à Identificação de Equipamentos Residenciais para Sistemas de Monitoramento não Intrusivo de Carga". Master's thesis, Universidade Federal do Pará, Belém, 2018.

[14] Y. Lu, S. Yi, N. Zeng, Y. Liu and Y. Zhang. "Identification of rice diseases using deep convolutional neural networks". *Neurocomputing*, vol. 267, pp. 378–384, December 2017.

[15] J. Redmon, S. Divvala, R. Girshick and A. Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016.

[16] C. Yu, S. Yang, W. Kim, J. Jung, K.-Y. Chung, S. W. Lee and B. Oh. "Acral melanoma detection using a convolutional neural network for dermoscopy images". *PLOS ONE*, vol. 13, no. 3, pp. e0193321, March 2018.

[17] J. Seetha and S. S. Raja. "Brain Tumor Classification Using Convolutional Neural Networks". *Biomedical and Pharmacology Journal*, vol. 11, no. 3, pp. 1457–1461, September 2018.

[18] F. Özyurt, E. Sert, E. Avci and E. Dogantekin. "Brain tumor detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy". *Measurement*, vol. 147, pp. 106830, December 2019.

[19] A. C. R. Marques, M. M. Raimundo, E. M. B. Cavalheiro, L. F. P. Salles, C. Lyra and F. J. V. Zuben. "Ant genera identification using an ensemble of convolutional neural networks". *PLOS ONE*, vol. 13, no. 1, pp. e0192011, January 2018.

[20] L. C. Ngugi, M. Abelwahab and M. Abo-Zahhad. "Recent advances in image processing techniques for automated leaf pest and disease recognition – A review". *Information Processing in Agriculture*, April 2020.

[21] X. Lin, X. Wang and Z. Hao. "Supervised learning in multilayer spiking neural networks with inner products of spike trains". *Neurocomputing*, vol. 237, pp. 59–70, May 2017.

[22] S. B. Jadhav, V. R. Udupi and S. B. Patil. "Identification of plant diseases using convolutional neural networks". *International Journal of Information Technology*, February 2020.

[23] V. Rezende, M. Costa, A. Santos and R. C. L. de Oliveira. "Aplicação de redes neurais convolucionais na classificação de doenças de plantas". 2019 CBIC XIV Congresso Brasileiro de Inteligência Computacional, 2019.

[24] V. Rezende, M. Costa, A. Santos and R. C. L. de Oliveira. "Image Processing with Convolutional Neural Networks for Classification of Plant Diseases". 2019 BRACIS 8th Brazilian Conference on Intelligent Systems, 2019.

[25] J. Deng, W. Dong, R. Socher, L. Li, Kai Li and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, June 2009.

[26] F. Chollet *et al.*. "Keras". Available in `https://keras.io`, accessed in 2020/07/09, 2015.

[27] B. T. P. and. "Acceleration of Stochastic Approximation by Averaging". *SIAM Journal on Control and Optimization*, vol. 30, no. 4, pp. 838–855, July 1992.

[28] S. Ruder. "An overview of gradient descent optimization algorithms". *CoRR*, vol. abs/1609.04747, 2016.

[29] A. J. Bell and T. J. Sejnowski. “The “independent components” of natural scenes are edge filters”. *Vision Research*, vol. 37, no. 23, pp. 3327–3338, December 1997.

[30] A. Kessy, A. Lewin and K. Strimmer. “Optimal Whitening and Decorrelation”. *The American Statistician*, vol. 72, no. 4, pp. 309–314, January 2018.

[31] S. Xie, R. Girshick, P. Dollár, Z. Tu and K. He. “Aggregated Residual Transformations for Deep Neural Networks”. 11 2017.

[32] J. Derrac, S. García, D. Molina and F. Herrera. “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms”. *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, March 2011.

[33] K. Doksum. “Robust Procedures for Some Linear Models with one Observation per Cell”. *The Annals of Mathematical Statistics*, vol. 38, no. 3, pp. 878–883, June 1967.